
Lasse Öberg

Linköping, Nov. 2007
Abstract

Wireless sensor networks are emerging from the mobile ad hoc network concept and as such they share many similarities. However, it is not the similarities that differentiates sensor networks from their ad hoc counterparts, it is the differences. One of the most important difference is that they should operate unattended for long periods of time. This is especially important since they usually rely on a finite energy source to function. To get this into a perspective, a sensor network constitutes of a sensor field where a number of sensor nodes are deployed. The sensor nodes relay the gathered information to a base station from which the data are forwarded either through a network or directly to the end-user. The communication between sensor nodes are conducted in an ad hoc manner, which means that paths toward the base station are dynamically constructed based on current network conditions. The network conditions changes and examples of this includes node failure, deactivated nodes, variations in the radio channel characteristics, etc.

As mentioned above, the sensor nodes are energy constrained and one of the more important design criteria is the life time of a sensor node or network. To be able to evaluate this criteria an energy dissipation model is needed. Most of the energy dissipation models developed for wireless sensor networks are not based on the basic sensor node architecture and as such they where not accurate enough for our needs. Thus, an energy dissipation model was developed. This model utilises the basic sensor node architecture to obtain the operation states available and their corresponding state transitions.

Communication is the most energy consuming task a sensor node can undertake. As such, the contributed energy dissipation model is used to evaluate this aspect of the proposed controlled flooding protocols. Generally, the controlled flooding protocols tries to minimise the number of forwarding nodes and by doing this they lower the energy consumed in the network. Along with this, the communication overhead of a protocol also needs to be taken into account. Our idea is to utilise the received signal strength directly to make forwarding decisions based on a cost function. This idea has a number of key features, which are: no additional overhead in the message, no neighbour knowledge and no location information are needed. The results from the proposed flooding protocols are promising as they have a lower number of forwarding nodes and a longer lifetime than the others.
Acknowledgement

There are many people that have helped me both before and during the writing of this licentiate thesis, to whom I would like to express my sincere gratitudes towards. First, I would like to acknowledge my supervisor Professor Youzhi Xu for his guidance, our discussions and especially for believing in my abilities.

I would also like to express my gratitude towards my examiner Professor Lars Wanhammar at Linköping University for enrolling me at the department of Electrical Engineering at the division of Electronic Systems.

In addition, I would like to express my gratitude towards Professor Thomas Ericsson at the division of Data Transmission at Linköping University for his ability to explain his courses in a very simple and understandable way.

I am grateful for the help and support provided by the members of the Electrical and Computer Engineering department at the School of Engineering and especially the members in the research area of Embedded Systems for our fruitful discussions. To mention a few of them: Per Karlsson, Shashi Kumar and Ragnar Nohre.

Finally, I would like to express my sincerest gratitudes towards my friends and family for their support and encouragement to finish this work and especially towards Emma for being who she is.

This work was supported and financed by the School of Engineering at Jönköping University.

Lasse Öberg
Jönköping, Nov. 2007
List of papers

Refereed papers included in this thesis.

I. Lasse Öberg and Youzhi Xu

II. Lasse Öberg and Youzhi Xu

III. Lasse Öberg and Youzhi Xu

IV. Lasse Öberg and Youzhi Xu

Refereed papers not included in this thesis.

- Lasse Öberg, Youzhi Xu and Ragnar Nohre

- Per Karlsson, Lasse Öberg and Youzhi Xu
Contents

1 Introduction .......................... 1
   1.1 Background .......................... 2
      1.1.1 Energy dissipation model ............. 2
      1.1.2 Efficient flooding ................... 3
   1.2 Contributions ...................... 5
   1.3 Thesis outline ..................... 5

2 Sensor Node Architecture ............ 7
   2.1 Basic sensor node subsystems ......... 7
      2.1.1 Processing subsystem ............ 8
      2.1.2 Sensor subsystem ................ 10
      2.1.3 Communication subsystem ......... 11
      2.1.4 Power-supply subsystem .......... 12
   2.2 Power management .................. 14
      2.2.1 Discrete operation states ........ 14
      2.2.2 Dynamic Voltage Scaling .......... 16
   2.3 Sensor Node Examples .............. 16
      2.3.1 MICAz mote ..................... 17
      2.3.2 BTnodes ......................... 17
      2.3.3 Zigbee-ready module ............. 17
      2.3.4 Sun Spot ........................ 18

3 Sensor Network Architecture ......... 19
   3.1 Communication approaches .......... 19
   3.2 Communication paradigms .......... 20
   3.3 Network model ..................... 22
      3.3.1 Design considerations .......... 23
   3.4 Protocol stacks ................... 25
      3.4.1 Open system interconnection model 25
      3.4.2 Sensor network protocol stack .... 28
   3.5 Medium access control ............. 29
      3.5.1 Wireless issues ................ 29
      3.5.2 Requirements .................. 30
6 Conclusion

6.1 Future work .................................................. 94
  6.1.1 Energy dissipation model ............................... 94
  6.1.2 Efficient flooding ...................................... 94
  6.1.3 Direction ............................................... 95
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Basic sensor node architecture.</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Energy overhead and saved energy due to a state transition.</td>
<td>16</td>
</tr>
<tr>
<td>2.3</td>
<td>Illustration of a) a non DVS system and b) a DVS system.</td>
<td>17</td>
</tr>
<tr>
<td>3.1</td>
<td>Illustrates the difference between a) unicast; b) broadcast; and c) multicast communication.</td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td>Difference between address and data centric paradigms.</td>
<td>21</td>
</tr>
<tr>
<td>3.3</td>
<td>A typical wireless sensor network.</td>
<td>23</td>
</tr>
<tr>
<td>3.4</td>
<td>The open system interconnection reference model.</td>
<td>26</td>
</tr>
<tr>
<td>3.5</td>
<td>Illustrates a sensor network protocol stack.</td>
<td>28</td>
</tr>
<tr>
<td>3.6</td>
<td>a) the hidden node problem and b) the exposed node problem.</td>
<td>31</td>
</tr>
<tr>
<td>3.7</td>
<td>MAC-protocol classification.</td>
<td>33</td>
</tr>
<tr>
<td>3.8</td>
<td>The ZigBee protocol stack.</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>Block diagram of the radio energy dissipation model.</td>
<td>42</td>
</tr>
<tr>
<td>4.2</td>
<td>Linear network model.</td>
<td>43</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparing direct versus multihop communication.</td>
<td>44</td>
</tr>
<tr>
<td>4.4</td>
<td>State transition diagram of the SEDM.</td>
<td>46</td>
</tr>
<tr>
<td>4.5</td>
<td>State transition diagram of the complete energy dissipation model.</td>
<td>48</td>
</tr>
<tr>
<td>4.6</td>
<td>Measured values for the scenario.</td>
<td>55</td>
</tr>
<tr>
<td>4.7</td>
<td>The sensor node topology for the environmental control scenario.</td>
<td>56</td>
</tr>
<tr>
<td>4.8</td>
<td>The active and inactive portion of the beacon interval.</td>
<td>57</td>
</tr>
<tr>
<td>4.9</td>
<td>The active portion of the beacon interval uses this superframe.</td>
<td>57</td>
</tr>
<tr>
<td>4.10</td>
<td>The inactive portion of the beacon interval is divided into ten sub intervals.</td>
<td>58</td>
</tr>
<tr>
<td>5.1</td>
<td>Black nodes form a dominating set and nodes with a square form a connected dominating set.</td>
<td>64</td>
</tr>
<tr>
<td>5.2</td>
<td>A categorisation for the flooding protocols.</td>
<td>66</td>
</tr>
<tr>
<td>5.3</td>
<td>Nodes within a radius of 2-hops from the source node.</td>
<td>68</td>
</tr>
<tr>
<td>5.4</td>
<td>Example of the iFSP protocol.</td>
<td>71</td>
</tr>
<tr>
<td>5.5</td>
<td>Example of SBA.</td>
<td>72</td>
</tr>
<tr>
<td>5.6</td>
<td>Node A is in a) inside the convex polygon and is in b) outside the convex polygon.</td>
<td>74</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.7</td>
<td>Example using the location-based scheme in a 4 node network.</td>
<td>75</td>
</tr>
<tr>
<td>5.8</td>
<td>a) Covering circles with hexagons. b) BPS for an ideal scenario.</td>
<td>76</td>
</tr>
<tr>
<td>5.9</td>
<td>Five node network topology used to illustrate how BPS works.</td>
<td>77</td>
</tr>
<tr>
<td>5.10</td>
<td>An example contour of equally received signal power when node $S$ transmits a message.</td>
<td>79</td>
</tr>
<tr>
<td>5.11</td>
<td>An example of the LQAF protocol using a five node network.</td>
<td>80</td>
</tr>
<tr>
<td>5.12</td>
<td>The rebroadcast area for the first redundant broadcast message using the RAF protocol. Thresholds $R_{Th,a}$ for node $a$ and $R_{Th,b}$ for node $b$.</td>
<td>82</td>
</tr>
<tr>
<td>5.13</td>
<td>An example of the RAF protocol using a four node network.</td>
<td>84</td>
</tr>
<tr>
<td>5.14</td>
<td>Sensor region with 200 nodes randomly deployed.</td>
<td>85</td>
</tr>
<tr>
<td>5.15</td>
<td>Protocol reachability.</td>
<td>87</td>
</tr>
<tr>
<td>5.16</td>
<td>Number of rebroadcast nodes.</td>
<td>88</td>
</tr>
<tr>
<td>5.17</td>
<td>Normalised number of dead nodes over time.</td>
<td>89</td>
</tr>
<tr>
<td>5.18</td>
<td>Reachability over time.</td>
<td>90</td>
</tr>
</tbody>
</table>
## List of Tables

2.1 Energy densities for some battery types. ................................. 13
2.2 Power densities for some power scavenging methods. ............... 14

4.1 The four operation states used in SEDM. ............................... 45
4.2 The four operation states used in this energy model for wireless sensor networks. $T_x = \text{transmit}, R_x = \text{receive}$ ................................. 47
4.3 The power consumption for the four operation states. ............... 52
4.4 Transition times both measured and obtained from data sheets. ... 53
4.5 Energy consumed in a transition for calculated and measured values. . 54
4.6 The calculated and measured energy consumption for the scenarios. . 56
List of Algorithms

1  Skeleton algorithm for controlled flooding protocols. 67
Chapter 1

Introduction

How people communicate with each other has evolved from simple speech to more sophisticated techniques such as telephone, e-mail, instant messaging, etc. All of these advanced techniques rely on communication networks to be able to forward the information to the appropriate users. In the beginning of these communication networks, wires were used as the main communication links and people used to decide which links that particular information needed to be forwarded onto. As the technology advanced wired communications in some areas has been replaced with wireless communication and people no longer needed to be part of the communication network.

In recent decades wireless communication technologies have seen an increased use and one of its driving forces is the reduced installation cost. Another one is its rapid deployment, where no or only partial infrastructure coverage exists for communication. Thus, the idea to form on the fly ad hoc networks emerged. An ad hoc network constitutes of communication devices that act as routers. These devices are moveable; they communicate with each other using wireless communication technologies; and they are battery-sustained. The communication between devices can occur even though they are not in communication range of each other. In that case they relay the information through other devices. The purpose of these types of network is for people to communicate with each other in geographic areas that have no or only partial infrastructure communication coverage.

Emerging from the typical ad-hoc networks is wireless sensor networks, which instead of human to human communication uses device to human communication. The general idea for the sensor nodes (very small communication devices fitted with sensors) is to gather information about their surrounding and forward it through other sensor nodes to a base station. From this base station an end-user retrieves the gathered information, which then can be viewed.
CHAPTER 1. INTRODUCTION

1.1 Background

In recent years wireless sensor networks have become an increasingly popular research area. The research in this area has led to the development of small sized, inexpensive and low-powered wireless sensor nodes. These sensor nodes are able to sense its surrounding, perform limited processing and communicate with nodes in its proximity using wireless communication technology. Since these sensor nodes have limited capabilities, the number of nodes deployed is usually in the order of hundreds or even thousands to provide a fault-tolerant and high quality-sensing environment. The collaboration between the sensor nodes takes the form of distributed collection and processing of data. In addition to fault-tolerant, the wireless sensor network also needs to be self-configuring and adaptive to its surroundings. This is due to the wide range of environments in which a fixed communication infrastructure is not available.

It is possible to deploy these wireless sensor networks in any type of environment imaginable such as remote geographic regions, office buildings, interior of planes and toxic urban environments. As such the application areas are numerous for these multipurpose networks. Examples of application areas for these wireless sensor networks include contamination tracking, habitat monitoring, traffic surveillance, health monitoring, automatic manufacturing, etc.

The material of this thesis constitutes of two different research areas. The main reason for this is that the future research direction will result in cluster construction and management algorithms, where the energy dissipation model will be used to evaluate these algorithms and the efficient flooding protocols will be used as building blocks in these algorithms. Below follows a brief description of the background and motivation for the two different research areas.

1.1.1 Energy dissipation model

Wireless sensor networks usually rely on energy from some type of battery and thus one of the main research objectives is to increase the life time of these networks. This can be accomplished by developing energy efficient micro-processors, sensors, I/O devices, communication protocols, etc. This can be further improved by utilising the behaviour characteristics of a wireless sensor node to predict when components need to be active or deactive without missing important events. This procedure by activating and deactivating components in real-time is called dynamic power management [13][17][56] and provides different power consumption levels, referred to as operation states of a wireless sensor node. Thus the life time of the sensor node and network as a whole can be increased. The difficulty is to accurately schedule the activation and deactivation of different components according to the behaviour of a sensor node, since the transition between two operation states both takes time and consumes power [62][61]. This power management scheduling [41] may also need to take other sensor nodes in the vicinity into account, since they might monitor the same area, require relaying of data, require position estimation, etc. To be able to optimise the dynamic
power management functionality [54] with respect to energy and delay constraints in a
wireless sensor node or network, an energy model that takes both of these constraints
into account for the operation states and the state transitions is needed.

An energy model should in our opinion provide developers and designers with the
means to construct energy efficient components and protocols for different applica-
tions. It is also important for the model to be able to evaluate how different compo-
nents interact with each other in a wide variety of application scenarios. For this to
be accomplished easily, the energy model should be both simple and adaptable to the
application needs.

The main motivation to develop our own energy dissipation model is based on the
need to evaluate the energy consumption of different protocols and algorithms in a
sensor network. Since this is a major bottleneck, if the sensor network shall be oper-
tional for a long period of time and at the same time be unattended. The protocols and
algorithms can for example be sleep scheduling algorithms, communication protocols,
location estimation algorithms, etc. This requires an energy dissipation model to incor-
porate a number of operation states related to the working conditions of a sensor node
and the corresponding state transitions. Since, most of the energy dissipation models
already developed for wireless sensor networks do not include all the components in a
sensor node and very rarely the state transitions [27][44]. We needed to develop our
own energy dissipation model. This energy dissipation model will be used in the future
to evaluate cluster construction and management algorithms that incorporates different
sleep scheduling policies.

1.1.2 Efficient flooding

A sensor network connects to the outside world using a sink (base station) through
which the end-user can send requests to the network and receive data from the net-
work. In some cases, the base station needs to propagate data to all sensor nodes in the
network. It can for example be software updates or requests concerning sensed infor-
mation. In other cases, a sensor node needs to notify nearby nodes concerning recently
collected data. This can for example be tracking assistance or propagation of sensor
readings. For sensor nodes to be able to relay this information in the network, both be-
tween sensor nodes and to or from the base station, a routing protocol is needed. In a
sensor network this is a challenging problem since the nodes are resource constrained
and there are a large number of nodes in the network.

Classic flooding is one of the simplest ways to relay data from one node to all other
nodes in a network. It is also used in some reactive routing protocols [52][32] mainly
in the pre-processing stage to find the intended destination, and in some cases the main-
tenance stage to repair the established route. However, classic flooding suffers from
disadvantages such as the broadcast storm problem [45]. A broadcast storm occurs
when many nodes close to each other in a geographic area retransmit a broadcast mes-
 sage approximately at the same time. This results in unnecessary collisions, contention
and redundancy in that geographic area and therefore increased energy consumption.
CHAPTER 1. INTRODUCTION

To overcome the broadcast storm problem, researchers have proposed techniques [64][53][69][75][70][59] to lower the number of retransmitting nodes in the network while attempting to maintain the number of nodes that receive the broadcast message, so called efficient flooding. For efficient flooding protocols to be used in wireless sensor networks there are some characteristics that need to be taken into account during the design of the protocol. These characteristics include:

A. **Scalability** is an important issue for wireless sensor networks since nodes are to be deployed in a large quantity. Thus, it is preferable that the protocol operates on local information rather than on global to limit the communication overhead.

B. **Memory** is also an important aspect due to rather limited availability on a sensor node. Therefore network protocols should have a low memory requirement.

C. **Energy efficiency** is a critical factor due to the limited energy supply in sensor nodes (i.e. usually battery-supplied). The communication is at the moment one of the most energy consuming operations a sensor node can undertake. Thus, reducing the number of transmissions and preventing nodes with low remaining energy to retransmit are important aspects to consider.

From the above characteristics centralised protocols are impractical to use in resource constrained wireless sensor networks. This is due to the large memory requirement needed and the large overhead to keep the network topology information up-to-date. The latter results in excess energy consumption. As such a more suitable approach is to use distributed protocols. These protocols can either rely on neighbour knowledge or use some other technique to decide the rebroadcast nodes.

The former (i.e. neighbour knowledge methods) protocols have some of the weaknesses seen in the centralised protocols, which are related to memory requirement (e.g. storage of neighbour tables) and communication overhead (e.g. beacon messages). But not to the same extent, as this knowledge only concerns the local topology.

The latter (i.e. non-neighbour knowledge methods) protocols have some advantages in that they are relatively simple, have a low communication overhead and a low memory requirement compared to the neighbour knowledge methods. However, the simplest schemes like the counter-based [45] and the probability-based schemes [45] are not as reliable in terms of their reachability and number of rebroadcast nodes as those schemes that rely on the nodes location. In the more advanced of these schemes the location information needs to be acquired and relayed. The relaying procedure usually adds this information to the broadcast message that is transmitted to the other nodes. Thus, our goal is to remove the need to add information to the broadcast message and still have a reliable, efficient and simple scheme.
1.2 Contributions

My research contributions are presented in Chapters 4 and 5. More precisely, it is the complete energy dissipation model \[48\]\[49\] and the measurements on a working sensor node in the former chapter. While, in the latter chapter it is the link quality-aided flooding protocol \[47\], the received signal strength-aided flooding protocol \[50\] and the protocol comparison. Below follows a short summary of the contributed material.

The complete energy model: This energy model is based on the basic sensor node architecture. From this architecture four useful operation states can be deduced, namely: the sleep, the sense, the transmit and the receive states. In addition to the operation states the state transitions are also included such that the scheduling between different operation states can be decided based on energy overhead. Furthermore, the measurements for the operation states and the state transitions are presented for a typical sensor node.

Link quality-aided flooding: The idea with this controlled flooding protocol is to utilise the received signal strength to obtain a link quality value. This link quality value is then used to make rebroadcast decisions based on a cost function. The protocol have a number of key features that are: no additional overhead in the message, no neighbour knowledge and no location information are needed.

Received signal strength-aided flooding: This protocol uses the same idea as that of the link quality-aided flooding protocol. However, instead of using the link quality value it uses the received signal strength to determine the rebroadcast order of the nodes using a cost function. This protocol also uses a threshold to adjust its reachability and it incorporates an energy aware metric that increases the time until a rebroadcast occurs for nodes with a low energy supply. The decision algorithm for the link quality-aided flooding protocol is a special case of this protocol and as such they share the same key features.

1.3 Thesis outline

This section discusses the remainder of the thesis and in which order it is intended to be read.

In Chapter 2 the basic sensor node architecture is introduced. Its subsystems are described and some commonly used power management techniques to lower the energy consumption are also explained. To give an idea of the type of components used in a sensor node, some commonly used components and sensor nodes are also presented. Thus, this chapter gives a fundamental overview of a sensor node and its basic architecture.

Following the overview of the sensor node the sensor network architecture is described in Chapter 3. The focus of this chapter is to give an overview of the communication structure in a wireless sensor network. This involves a communication
paradigm, a rather unique communication approach, which can be looked at as reverse multicast, and a network model that is different to the one used in traditional networks. Furthermore, this chapter also describes a network protocol stack and the problems related to the medium access control.

Chapter 4 describes three energy models that have been used to evaluate different protocols for wireless sensor networks. In addition, to these energy models a power management model is described and measured values are presented for a typical sensor node. These measurements include the power consumed in each operation state and the state transitions for the proposed energy dissipation model. Furthermore, the time it takes to complete each state transition is also included. Using these measurements the complete energy dissipation model is compared to that of a working sensor node.

In Chapter 5 classic flooding and controlled flooding protocols are described. The latter is divided into two classes of methods: neighbour knowledge and non-neighbour knowledge methods. In each of these classes, two protocols are described in more detail. In addition to the protocols reviewed, two non-neighbour knowledge protocols are proposed and these are the link quality-aided flooding and the received signal strength-aided flooding protocols. Both of these mentioned protocols rely on the received signal strength for their rebroadcast decisions, which is a unique variation of existing techniques. Four of the previously described protocols are then compared to each other using a network simulator.

Finally, Chapter 6 gives a summery of the concluding remarks presented in Chapters 4 and 5. Furthermore, a direction for future work related to the communication aspects of this thesis are presented.

Readers familiar to the hardware design of a sensor node may skip Chapter 2 since it presents the preliminaries for Chapter 4 and readers familiar with the communication aspects in wireless sensor networks may skip Chapter 3 since it presents the preliminaries for Chapter 5. From the above description Chapters 4 and 5 can be read independently.
Chapter 2

Sensor Node Architecture

In this chapter a brief description of the architecture of wireless sensor nodes are presented. The description includes the basic sensor node architecture, some commonly used dynamic power management methods and examples of sensor nodes. The subsystems included are also described and both positive and negative aspects are put forth. How these subsystems functionality can be utilised to improve the lifetime of a sensor node or network (i.e. to lower the power consumption based on actual working conditions) is also presented. Two techniques are presented that accomplish this and these are discrete operations states and dynamic voltage scaling. These techniques are usually deployed together. To get an idea about the components used in today’s sensor nodes, some practical sensor node examples are presented.

2.1 Basic sensor node subsystems

In the hardware design process of sensor nodes, the application requirements play a critical role in the choice of hardware components with regard to size, cost and energy consumption. The size constraint limits the available space for components and both the energy and cost constraints limits the types of components that can be used. As such, these constraints provide a limit on a sensor nodes capability with regard to available energy, processing power, storage capabilities, communication resources, etc.

A sensor node consists of a minimum of four subsystems which is referred to as the basic sensor node architecture [68][57][9]. This architecture is designed to provide the sensor node with the basic means to communicate with nodes in its proximity, process data, gather data and to work unattended for long periods of time in a wide variety of environments. As such, this architecture consists of the following four subsystems:

A. A processing subsystem consisting of a controller and a memory unit.

B. A sensing subsystem with its associated sensors and analog-to-digital converters (ADC).
CHAPTER 2. SENSOR NODE ARCHITECTURE

Figure 2.1. Basic sensor node architecture.

C. A communication subsystem with a transceiver such as a radio frequency or an acoustic transceiver.

D. A power supply subsystem that provides energy to the other subsystems.

The basic sensor node architecture is shown in Figure 2.1. All these subsystems are then controlled by the operating system, running on the controller, using the appropriate drivers, protocols and algorithms. The basic sensor node architecture can be extended by adding additional subsystems which provide the sensor node with added capabilities like location awareness, mobility, energy replenishing systems, etc. The four subsystems are described below according to the order in the list above.

2.1.1 Processing subsystem

The processing system usually constitutes of a controller and a memory unit. The controller unit can either be a micro-controller unit (MCU), a field programmable gate array (FPGA) or an application specific integrated circuit (ASIC). However, in this work we assume a MCU based node architecture. Below follows a short descriptions of the micro-controller and the memory units of the processing subsystem.

Micro-controller

The MCU provide the sensor node with something that can be seen as intelligence, in the sense that it controls other components using the appropriate drivers, algorithms and protocols. Basically the MCU decides from which internal source the data shall be collected from, how the data should be processed (e.g. data aggregation, sorting, comparing) and in which state each component needs to be assigned to (e.g. sleep, active, idle). It also decides, based on processing results, queries, etc, how data should
be distributed in the network and if other nodes can relay data using this particular node.

The different tasks, mentioned above, that are run on the MCU are all software controlled. The use of software which is easily altered or replaced depending on the circumstances makes the MCU a very flexible controller. In addition, to the ability to easily alter or replace the software on the MCU, it usually has a large number of external interfaces that can be connected to other components (e.g. Two Wire Interface, Serial Peripheral Interface) and support to handle time critical tasks (e.g. communication protocol execution). The MCU also has the ability to assign to itself a number of operating states in which parts of the MCU, for example, can be turned off. Furthermore, it can use dynamic voltage scaling to alter the operating frequency to fit the application needs.

However the flexibility that the MCU provides comes at a cost, with typical trade-offs towards energy-efficiency and performance loses. A solution that can provide a higher energy-efficiency and improved performance with a loss towards the flexibility that the MCU provide is the use of an ASIC, in which everything is constructed in hardware.

Memory

The memory that is used can be divided into two different categories, random access memory (RAM) and read-only memory (ROM) types. The RAM provides short term storage for packets from other nodes, intermediate sensor readings, working data, etc. The main advantage with RAM is that it is fast and its disadvantage is that the content of the memory is lost when power is removed. One commonly used RAM type in a MCU is static RAM, the term static means that it retains its content until power is removed. This is unlike dynamic RAM which needs to be periodically refreshed. The ROM types are usually used for long term storage of data, programs, etc. The ROM usually used is the electrically erasable programmable read only memory (EEPROM) and flash memory. Flash memory is a type of EEPROM that allows writing of blocks of data instead of one byte at a time. It can also be used for intermediate data storage when power needs to be removed from the RAM or when there is not sufficient space available in the RAM.

Hardware examples

For the hardware examples in the processing subsystem I choose to focus on the micro-controllers rather than the memory. Thus, below follows a short description of two commonly used embedded processors that are used in sensor nodes today. These are the Atmel ATmega processors and the family of MSP430 processors from Texas Instrument.

**Atmel ATmega:** The Atmel ATmega 128L is a low power 8*bit* micro-controller in the AVR series. It uses a RISC (reduced instruction set computing) core running
CHAPTER 2. SENSOR NODE ARCHITECTURE

at maximum 8 MHz. Some of its features are: six different sleep modes used for power saving; it is equipped with external interfaces for common peripherals (e.g. byte oriented two wire serial interface, master/slave SPI interface.); and it incorporates a real time counter and two extended 16 bit timer/counter each with separate prescalars. This micro-controller is for example used in the BTnodes [4], MICAz node [6] and the Zigbee-ready module [39].

**Texas instruments MSP430:** MSP430 is a family of ultra low power micro-controllers from Texas instrument. These micro-controllers use a 16 bit RISC core running at maximum 16 MHz. This micro-controller family comes with a wide range of possible interconnections and an instructions set that allows the use of different kinds of peripherals easily. This family of micro-controllers is for example used in EYES nodes [5] and in ScatterWeb [7].

### 2.1.2 Sensor subsystem

The sensor subsystem constitutes of a number of sensors that collects data for further processing by the micro-controller. The sensors are either analog or digital devices and it depends on how its output is produced. An analog sensor must be connected to ADC before the data can be processed, which is not the case for a digital sensor. These sensor devices can be further categorised as active or passive sensor. This mainly impacts the amount of energy that is consumed during operation.

**Active sensors:** This category actively explores the environment by sending out pulses both directional and omnidirectional. Sensors in this category include radars, sonars and some types of seismic sensors. These sensors usually consume a large amount of energy due to the pulses needs to propagate into the surrounding environment.

**Passive sensors:** This category of sensors gathers data that is available at the sensor. This means that it do not actively searches for a phenomenon, but wait until the phenomenon reaches or propagates to the sensor. This makes passive sensors very energy efficient as there is no active component that continuously searches for a phenomenon. Some of these passive sensors are self powered in the sense that the phenomenon generates the needed power. However, power is still needed to amplify and convert the obtained signal. Typical examples of passive sensors are temperature, humidity, magnetic, light, vibrations sensors, etc.

Directional sensors in both the active and passive category might need a repositioning system to adjust the direction of the sensor. This type of system adds an extra energy consuming component to the system, which in some cases makes it better to have more sensors covering the entire environment than using a repositioning system with a single sensor.


2.1.3 Communication subsystem

The communication subsystems responsibility is to provide the sensor node with communication capabilities, in such a way that the sensor nodes are able to communicate with other sensor nodes in its proximity. The communication medium for the sensor node varies and depends on the actual application. In some cases wired communication can be applied, but its applications are limited. A much more interesting technique to use in this subsystem is a wireless communication component. The wireless communication medium used for sensor networks includes radio frequency, optical and acoustic (e.g. ultrasound) techniques [68]. The most common choice, when constructing a sensor node today, is to use a radio frequency (RF) transceiver. This is due to the fact that it fits the requirements of most wireless sensor network applications. Below follows a description of the mentioned wireless communication techniques.

**Radio frequency:** RF-based [68] communication provides long range and high bitrate communication with acceptable error rates. At the same time the sender does not need to be in line of sight of the receiver. These aspects of RF-based communication make it an interesting technology for use in wireless sensor networks. However, one complication that arises by using this technique is its rather high energy consumption.

**Optical:** Optical based [68] communication provides low energy consumption per bit for both detection and generation of the optical light. Its construction is both simpler and smaller than the corresponding RF-based counterpart and provides concurrent communication with negligible interference. However, by using this technique both the sender and the receiver need to be in line of sight of each other for communication. This limits the possible placement of sensor nodes and in an outdoor environment it is heavily influenced by the weather conditions.

**Acoustic:** Acoustic based communication is not very attractive to use in open air environments partly due to the large Doppler shifts generated compared to other alternatives and the reverberation. The reverberation is the lingering of sound in for example a room when the sound source has ceased producing. However, acoustic based communication has its uses in communication mediums, where the waves for both optical and RF based communication does not penetrate the surrounding medium very well. One such communication medium is water in which the water acts as a conductor for the electromagnetic waves used in RF based communication. This in turn decays the signal very rapidly and the decay depends on the transmitted carrier frequency. The higher the carrier frequency the faster the transmitted signal decays.

**Wake-up radio**

The transceiver is one of the most power consuming components of a sensor node when it comes to listen to the channel, receive or transmit data. The two communication
states transmit and receive, are both important since it allows the sensor node to relay its and other nodes data. But waiting to receive data, by listening to the channel, is not an energy efficient approach, since the receiver circuitry must be active. Thus it would be an attractive choice to be able to turn the transceiver off during times when no data is being received or transmitted.

The idea, with a wake-up radio [24][55][19][76][43], is to have a specialised circuit that detects when a packet is inbound and notifies the appropriate component which in turn schedules the activation of the main transceiver so it is able to receive the incoming data. The goal with such a wake-up receiver is to have very low power consumption, preferably less than $1 \mu W$, and to be able to distinguish the destination address, so that the main receiver is only activated for packets that are addressed to this sensor node. Even without the use of address recognition this approach lowers the power consumed by the transceiver enormously. Since, the transmitter spends most of its active time either transmitting or receiving data and not listening to the channel. So far, constructing a wake-up radio that are able to distinguish the destination address has not been achieved.

**Hardware example**

The discussion on the communication subsystem is concluded by a short description of one commonly used RF-transceiver today, namely Chipcons CC2420. Most of the used communication subsystems are off-the-shelf products with different advantages and disadvantages and as such no optimal choice is available.

**Chipcon CC2420:** Chipcon’s CC24020 operates at the $2.4 \ GHz$ frequency band with a built-in physical-layer and parts of the MAC-layer. It uses the IEEE 802.15.4 standard. Thus, this transceiver requires that the implemented MAC-layer in the MCU is compatible with this standard. Other consequence of using this standard is that the transceiver uses a DSSS (direct-sequence spread spectrum) baseband modem with an effective data rate of $250 \ kbps$. An example of a sensor node that uses this RF-transceiver is the zigbee-ready module [39].

**2.1.4 Power-supply subsystem**

The power-supply subsystem provides power to the other components of the sensor node and there exist a number of different methods in which this can be accomplished. These methods can be classified into the following categorises [33]: power distribution; energy reservoirs; and power scavenging. The former category is not of particular interest, since it includes techniques (e.g. radio frequency, radiation, wires, acoustic emitters and light) that are not appropriate for the majority of wireless sensor networks. The other two techniques are described in more detail below.
2.1. BASIC SENSOR NODE SUBSYSTEMS

Table 2.1. Energy densities for some battery types.

<table>
<thead>
<tr>
<th>Energy source</th>
<th>Energy density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery (Lithium)</td>
<td>2880 J/cm³</td>
</tr>
<tr>
<td>Battery (Lithium, Rechargeable)</td>
<td>1080 J/cm³</td>
</tr>
</tbody>
</table>

Energy reservoirs

In this technique the energy is stored in some type of container and as energy is consumed the available energy is decreased. The most commonly used of these techniques are the battery, which is the usual way of powering a sensor node today. In Table 2.1 the energy densities for some batteries are presented [34]. The batteries can either be non-rechargeable, so called primary batteries, or rechargeable, so called secondary batteries. If rechargeable batteries are used power scavenging devices can be used to restore energy into the battery. One thing that needs to be considered when charging batteries is that extra electronics is often needed to control the charging profile. Other techniques in this category includes: micro-fuel cells, ultra capacitors, micro heat engines and radioactive power sources. However, the current sizes of these techniques are an issue if they are going to be used in sensor nodes today.

The batteries usually have a typical problem and that is the reduction of voltage as its capacity drops and influence the operation cycle of a sensor node. The consequences include frequency oscillation which can lead to unstable sensor node behaviour. This problem can be overcome by regulating the voltage delivered to the sensor nodes circuitry. When the supply voltage drops the regulator draws an increasingly higher current from its source and thus reduces the time until the energy source is empty. DC-DC converter and voltage regulator are circuits that can provide a constant voltage to the sensor node. These kinds of circuits also consume energy for their operation and thus reducing the efficiency. The advantages of predictable operation by using this type of circuit, during node operation, might outweigh its disadvantage of additional energy consumption.

Power scavenging

This technique helps sensor nodes and networks increase their operation cycle by utilising its surrounding environment to generate power for the sensor node. There are several different approaches that exist today for power scavenging, some of these are: photovoltaic e.g. solar cells; temperature gradients; air/liquid flows; pressure variations; and vibrations. In Table 2.2 the power densities for some power scavenging methods are presented [34].

There exists at least one problem with these techniques. It is due to the fact that these power sources have a hard time providing a continuous power level for the duration of the sensor node lifetime. This is because of environmental changes surrounding the sensor node. Due to this problem, power scavenging methods are usually combined
Table 2.2. Power densities for some power scavenging methods.

<table>
<thead>
<tr>
<th>Energy source</th>
<th>Power density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solar (outdoor, direct sunlight)</td>
<td>15 mW/cm²</td>
</tr>
<tr>
<td>Solar (outdoor, cloudy day)</td>
<td>0.15 mW/cm²</td>
</tr>
<tr>
<td>Vibrations</td>
<td>0.01 - 0.1 mW/cm²</td>
</tr>
<tr>
<td>Acoustic noise</td>
<td>3 \cdot 10^{-6} mW/cm² at 75 dB</td>
</tr>
</tbody>
</table>

with rechargeable energy reservoirs (e.g. batteries). This in turn, as mention above, requires additional circuitry for recharging, which by itself consumes additional energy and requires a reservoir technology that can be recharged at low currents. Another technique mentioned in the literature [33], is to analyse the energy scavenging characteristics. Based on this knowledge the algorithm adapts the task execution pattern of a sensor node to fit the obtained knowledge as closely as possible.

2.2 Power management

Power management is a set of techniques used in some electronic appliances to reduce the power waste. These techniques usually put the system into a low power state or turn it off when it has been inactive for a period of time. A more advanced technique is referred to as dynamic power management (DPM) [14][12][42]. Basically, a system that utilises DPM functionality is a system in which the power consumption characteristics is adjusted in real time to minimise power waste, but still meets its performance requirements. DPM includes techniques such as dynamic voltage scaling [40][74][73], dynamic process and temperature compensation and idle time prediction for controlling low-power modes (e.g. sleep, doze, etc.). This section highlights some of the consideration that needs to be taken into account during a transition between different operation states when idle time prediction is going to be used. It also describes the procedure of dynamic voltage scaling and how this can reduce the power waste in a system.

2.2.1 Discrete operation states

For all components in a subsystem there exist a number of predefined states in which the power consumption and the number of active parts are differentiated. In each component, there are at least two such states that can be used, these states are on and off. However, many components available today provide more than these two states, in such case the state on is divided into a number of additional states. All states available in a component can be seen as an ordered set, \( S \), where the order of the states \( s_i \in S \) depend on an increasing power consumption level. Take for example a component with four different states. This corresponds to the following set \( S = \{s_{\text{off}}, s_1, s_2, s_{\text{active}}\} \). Each of these states, left to right, has an increased power consumption and as such the
active parts in this subsystem are differentiated. The idea of using different operation states is to reduce the power waste by switching between the available states in \( S \). This is done when the requirements of this particular subsystem can either not be met or another state can meet the requirements with less power waste.

If we look at a RF-transceiver, which in this case have four states, namely: off; idle; receive; and transmit. This corresponds to the ordered set \( S = \{ s_{\text{off}}, s_{\text{idle}}, s_{\text{rx}}, s_{\text{tx}} \} \). The states \( s_{\text{tx}} \) and \( s_{\text{rx}} \) transmit and receive data. Both of these states are active states, i.e. they are doing something constructive with the resources that are provided to this component. In comparison to these active states, the state \( s_{\text{idle}} \) can be seen as a waiting state where the transceiver waits for a transition to the states \( s_{\text{tx}}, s_{\text{rx}} \) or \( s_{\text{off}} \). It would be preferable to transition the RF-transceiver to the state \( s_{\text{off}} \) instead of the state \( s_{\text{idle}} \) after transmitting or receiving data. This is true if no scheduled transmission or receptions of data will occur in the near future. In this case, by lowering the time spent in the state \( s_{\text{idle}} \), less power is also consumed.

However, the problem is that a state transition is not instantaneous and costs both power and time, in which the component is waiting for the transition to complete. This complicates the use of discrete operation states and sometimes makes it worthwhile to stay in a higher power state. Since the energy overhead for a state transition can be higher than spending the time in the high power state. Figure 2.2 shows the power and latency overhead for a state transition [62]. At time \( t_1 \) a node finishes its processing of the current event and the next event arrival is expected at time \( t_2 \). Then, at time \( t_1 \), a decision needs to be made whether the component should stay in the current state \( s_{\text{active}} \) with a power consumption of \( P_{\text{active}} \), or conduct a state transition to the low power state \( s_{\text{sleep}} \) with a power consumption of \( P_{\text{sleep}} \). Assume that the transition time from \( s_{\text{active}} \) to \( s_{\text{sleep}} \) is \( \tau_{\text{down}} \), the transition time from \( s_{\text{sleep}} \) to \( s_{\text{active}} \) is \( \tau_{\text{up}} \), and that the power consumed during a transition is \( (P_{\text{active}} + P_{\text{sleep}}) / 2 \), which is a simplification of the actual power consumed during a transition. The energy saved between these events is

\[
E_{\text{save}} = P_{\text{active}}(t_2 - t_1) - P_{\text{sleep}}(t_2 - t_1 - \tau_{\text{down}}) - \frac{P_{\text{active}} + P_{\text{sleep}}}{2}\tau_{\text{down}} \quad (2.1)
\]

When the new event finally arrives, it results in an additional overhead before the event can be processed. During this transition it is assumed that no productive work can be done. Thus, the energy overhead is

\[
E_{\text{overhead}} = \frac{P_{\text{active}} + P_{\text{sleep}}}{2}\tau_{\text{up}} \quad (2.2)
\]

The decision whether or not to change state at time \( t_1 \) is straight forward and leads to that a transition is beneficial only if \( E_{\text{save}} > E_{\text{overhead}} \), otherwise a transition is not commenced. Since the consumed energy is dependent on the time, a state transition is beneficial if the time to the next event arrival is large enough.
2.2.2 Dynamic Voltage Scaling

Components without a DVS system operate at a fixed performance level and complete the assigned task as fast as possible. In most cases this is before their deadline has been reached and then the component enters a low power mode for the remainder of the time. This operation is illustrated in Figure 2.3 a).

On the other hand, components with a DVS system can adjust their operating frequency in such a way that the task is finished just in time, i.e. as short time before the deadline as possible. This feature is illustrated in Figure 2.3 b). The reduction of the operating frequency $f$ allows for a reduction of the supply voltage $V_{dd}$. This results in a reduced power consumption as shown by equations (2.3) and (2.4) below.

\begin{align}
\text{Power} & \propto f V_{dd}^2 \\
\text{Energy} & \propto V_{dd}^2
\end{align}

However, the reduced operating frequency also results in a linear increase in the time to complete a task, which leads to a squared energy reduction that depends on the supply voltage. Thus, using DVS is a very efficient approach to reduce the energy consumption in a component. This is especially true if the workload of a component varies very much. Though, it is important to remember that the control of a DVS system requires knowledge about the execution time of each task, which in turn may need sophisticated prediction mechanisms to set the correct voltage and frequency.

2.3 Sensor Node Examples

In recent years there are many universities that have started research in the area of wireless sensor networks, some of these universities uses sensor nodes developed by
other schools or companies. But many still develop their own sensor node for use in their applications. Below follows a short description of some of the sensor nodes that are available today. Other examples include: EYES node [5]; and Scatterweb [7].

### 2.3.1 MICAz mote

The Mica motes are initially developed by the Berkley University collaborating with Intel. These nodes are now commercially available through the company Crossbow. One of the more interesting motes in the Mica series is the MICAz [6] that uses TinyOS as its default operating system. The significant hardware components used are: the ATmega 128L micro-controller developed by Atmel, an IEEE 802.15.4 compliant RF-transceiver and connectors such that additional components can easily be added.

### 2.3.2 BTnodes

The BTnode platform [4] was developed at ETH Zürich as collaboration between the computer engineering networks laboratory (TIK) and the research group for distributed systems. This is to serve as a research and demonstration platform for mobile and ad-hoc connected networks, which includes sensor networks. The significant hardware components used in the BTnodes are: Atmels micro-controller ATmega 128L, a bluetooth capable transceiver and Chipcons CC1000 transceiver.

### 2.3.3 Zigbee-ready module

The zigbee-ready module (ZRM) [39] has been developed as a master thesis work at Linköping university. Its intended application areas do not include long term operation.
and as such the designers have not chosen components based on this criterion. The main board consists of the following significant components: the ATmega 128L MCU developed by Atmel; and the Chipcon CC2420 RF-transceiver. The functionality of the main board can easily be extended by connecting so called secondary boards that add functionality to the main board.

### 2.3.4 Sun Spot

The sun spot sensor node [8] is developed by Sun Microsystems and utilises the Java 2 micro edition virtual machine. The virtual machine is run directly on the processor without the need for an operating system. This is the first sensor node that can be fully controlled by the use of the Java programming language. However, this is also one of the most powerful sensor nodes with respect to processing power. It includes an ARM920T MCU and an IEEE 802.15.4 compliant RF-transceiver.
Chapter 3

Sensor Network Architecture

In this chapter we describe the communication aspects of a wireless sensor network. This relates to how the nodes in these networks cooperate with each other and the unique properties compared to other networks. This chapter begins to describe the data delivery techniques commonly used in other networks and then continues to present two different communication paradigms. One of these paradigms is more targeted towards traditional networks and the other one is more adapted towards wireless sensor networks. After these descriptions the sensor network model is described. This includes the different building blocks used in a wireless sensor network, the communication categorise within these types of networks, the data delivery requirements and the goals. From the previous descriptions some important design considerations are put forth. For different sensor networks to be interoperable a common protocol stack is needed that defines how the protocols should behave. Thus a description about a general and a wireless sensor network protocol stack is presented. The last part of this chapter is focused on the description of the medium access control related to its issues, requirements and categorise.

3.1 Communication approaches

By approaches of communication I mean the three different data delivery terms commonly used in communication networks today. These are unicast, broadcast and multicast. The two foremost terms are each others opposite and can be seen as the two extreme cases. And the latter term describes the communication term that is the middle way between these two extremes. The three terms are described in more detail below.

Unicast: The term unicast describes the delivery of data that occurs from one source node to one destination node in a network, see Figure 3.1 a). This is the most common technique used in data networks today.

Example of unicast techniques include protocols such as: the hypertext transfer protocol (http), file transfer protocol (ftp), etc.
CHAPTER 3. SENSOR NETWORK ARCHITECTURE

Figure 3.1. Illustrates the difference between a) unicast; b) broadcast; and c) multicast communication.

Broadcast: Broadcast is the complete opposite of unicast and describes the communication in which the delivery of data occurs from one source node to all possible destinations in a network, see Figure 3.1 b). In practice the scope of a broadcast is limited in some way to reduce the overhead that otherwise is generated. Example of techniques that can utilise broadcast communication includes: on-demand routing protocols, cluster formation protocols, etc.

Multicast: Multicast communication can be seen as the middle way between the two extremes of unicast and broadcast communication. It describes the delivery of data from one source node to a group of destination nodes in a network, see Figure 3.1 c). The goal of this communication approach is to use the most efficient strategy to deliver data using as few links as possible. Basically this means that a multicast tree is constructed with the source node as a root and the receiving nodes as leaf nodes. Notice that if the number of receivers is one, it is referred to as unicast; and if all nodes in the network are the intended receivers, it is referred to as broadcast. However, the links chosen for delivery of data are optimised using the multicast approach in comparison to the broadcast approach. Example of multicast applications include: streaming of multimedia content, multiplayer games, etc.

3.2 Communication paradigms

There is a fundamental difference between the communication paradigms [36] of wireless sensor networks and traditional networks even though they share some similarities. These differences involve both routing and application requirements. One difference
3.2. COMMUNICATION PARADIGMS

between these types of networks is that in a sensor network data is usually relayed from nodes in the network towards a base-station or sink. This can be seen as a reverse multicast, rather than communication between any pair of nodes that usually is the case in traditional networks. Another difference is that data is collected by multiple sensors that are working together in proximity of a phenomenon. This means that the relayed data can contain redundant information and this redundancy can be exploited by the use of a data aggregation technique. Above some of the differences between the communication of traditional and sensor networks are highlighted and these differences have shown the way from an address centric communication perspective to a data centric communication perspective. Below follows a short description of both the address and the data centric communication paradigms.

**Address centric:** In address centric communication particular nodes are uniquely defined using addresses. This is done in such a way that the transfer of data can occur between two or more devices. The protocols developed using the address centric paradigm does not care about the content transferred. They only consider the available paths between source node and the destination node/nodes.

**Data centric:** Using the data centric communication paradigm, basically means that the content transferred is of more importance than the node transmitting the data. In sensor networks this paradigm is useful since nodes are usually deployed redundantly to have a better protection for node failures and to compensate for low quality sensor readings. This can generate a lot of extra data to be propagated towards the sink (base-station). Thus, to be able to fully utilise the nature of the data centric communication paradigm it is often combined with a data aggregation technique. This to reduce the data propagated in the network.

This example illustrates the difference between address and data centric routing schemes [72]. In Figure 3.2 a) the address centric scheme is shown and it uses the
shortest path between source and destination pairs to route data. Thus, the source node $S_1$ routes data 1 through the intermediate nodes $A$ and $B$ to the sink node and source node $S_2$ routes data 2 using node $C$ as a relay to the sink node. In comparison, the data centric scheme tries to combine the data whenever possible, as shown in Figure 3.2 b). This is illustrated by that the source node $S_1$ tries to routes data 1 through node $A$ to source node $S_2$. At node $S_2$, data 1 and data 2 is combined using a data aggregation technique. When this is accomplished, data $1 + 2$ is routed through the intermediate node $C$ to the sink node. In these two cases, the number of transmissions needed to route data from the source nodes to the sink node are: five for the address centric scheme; and four for the data centric scheme. This illustrates the possible gains that can be obtained using a data centric communication paradigm compared to the address centric paradigm in wireless sensor networks.

3.3 Network model

A typical sensor network [9] consists of a number of sensor nodes that are distributed in a sensor field i.e. the geographic region that is going to be monitored. These sensor nodes are able to communicate with other nodes in its proximity using wireless communication technology and to collect data from a phenomenon or a set of phenomena. The phenomenon is the entity of interest to the end-user. Thus the collected data are forwarded using other sensor nodes to a sink node. The sink node is a form of collector/informer that sends end-user interests (e.g. queries) to the sensor network and collects data related to these end-user interests from the network. The data of interest for a particular end-user is either available directly to the end-user or it must be forwarded through a network before it is available to the end-user. This is illustrated in Figure 3.3. The sink node, network and end-user are sometimes referred to as the observer of a phenomenon or a set of phenomena.

The communication within a sensor network, mainly between sensor nodes, can conceptually be divided into two categorise: application and infrastructure communication [66]. The former relates to the transfer of sensed data or information related to that data. The goal of the transfer is to relay data in the network to the sink node about the phenomenon of interest. There are two distinct models in this category, one is the cooperative and the other is the non-cooperative transfer of data. The cooperative is basically the data-centric paradigm and the non-cooperative is the address-centric paradigm. The infrastructure communication refers to the communication needed to configure, maintain and optimise the operation of a sensor network. This is due to the ad-hoc nature of sensor networks, the possibility of node failures, sensor nodes in power save mode, etc. Thus the infrastructure communication is needed to keep the network functional, ensure robust operation in dynamic environments and try to optimise overall performance.

The data delivery requirement, which is the interest an end-user has in a sensor network, can be classified into four categorise: periodic, event-driven, observer-initiated
3.3. NETWORK MODEL

A typical wireless sensor network.

Figure 3.3. A typical wireless sensor network.

and hybrid [66]. This is basically the traffic generation requirement of the sensor application. Using the periodic data model, the sensor node relays data at a specific periodicity. One extreme in this class is to continuously transmit data at a specific rate, which is when the time interval is close to zero seconds. In the event driven data model, a sensor node initiates communication only if a predefined event occurs. Here, the end-user is only interested in the occurrence of a phenomenon or a set of phenomenons. In the observer-initiated model, a sensor node only reply to a request or query from the end-user that concerns a phenomenon or a set of phenomenons. Two or more of these approaches can coexist in the same sensor network and this result in the hybrid model.

The goal of a sensor network is to reliably relay data of a phenomenon or a set of phenomenons that are of interest to the end-user. This, for the sensor network, includes creating and maintaining reliable paths and/or multiple paths from the phenomenon to the sink. This must be accomplished under dynamic conditions while meeting the applications requirements (e.g. low latency, low energy, high accuracy and fault tolerance). Dynamic conditions that can occur in a sensor network includes, but are not limited to: mobility for phenomenon, sensor nodes and sink nodes; sensor node failure; change in end-user interests; and variations in transmission distance due to signal attenuation, multipath fading, interference, etc.

3.3.1 Design considerations

The design considerations for wireless sensor networks are very similar to those of ad-hoc networks. But the focuses for those considerations are very different and as such
CHAPTER 3. SENSOR NETWORK ARCHITECTURE

needs to be emphasized. The considerations that are focused in wireless sensor networks [66][9] are fault tolerance, energy efficiency, latency, accuracy and scalability. These considerations are further explained below.

**Fault tolerance:** In a sensor network individual sensor nodes may fail due to a number of reasons. This can for example be lack of power, physical damage and interference (e.g. jamming, environmental, etc.). Even due to these reasons the sensor network must remain fully operational from the applications point of view. This means that the network conditions should be transparent to the end-user and that the end-user receives its interests within the given requirements.

The fault tolerance $R_k(t)$ has been modeled in [29] by using the Poisson distribution to obtain the probability of no failure occurrence for a sensor node in a given time frame $(0, t)$:

$$R_k(t) = e^{-t\lambda_k},$$

where $\lambda_k$ is the failure rate of sensor node $k$ and $t$ is the time period.

**Energy efficiency:** Since sensor nodes can only be equipped with a limited power source an important aspect is the lifetime of a sensor network. This is due to the roles a sensor node has in a sensor network, it must act as a data collector and route data collected by other nodes in the network towards the sink. This drains power from the power source rather quickly, especially during communication. Thus, power conservation and power management are important design aspect of protocols and algorithms for a sensor node and networks.

**Latency:** Usually the end-user is interested in knowing about a phenomenon within a given time frame i.e. delay requirement. For communication, the latency is the time difference between the occurrences of a transmission of data related to a phenomenon from a sensor node until the observer receives that data. In some applications the latency requirement can be subjected to more than one sensor node that needs to report a phenomenon within a given latency.

**Accuracy:** The application requirements determine the accuracy of the gathered data about a phenomenon or a set of phenomenons. This is the primary objective of the observer. It is important to notice that there is a trade-off between accuracy, energy efficiency and latency. As such, the infrastructure provided by the sensor network should adapt to the application requirements so that the given accuracy and latency is obtained with minimal energy consumption.

**Scalability:** Depending on the application requirements (i.e. accuracy, fault tolerance, etc.), the number of sensor nodes distributed inside a sensor region can be in the order of hundreds or even thousands of nodes. Thus, the schemes developed for wireless sensor networks must be able to cope with this magnitude of nodes in close proximity to each other and to utilise the high density in the sensor region.
3.4 Protocol stacks

Two protocols stacks are presented, the first is a general protocol stack that tries to include the needs for all types of communication networks and the second is a protocol stack adapted specifically for wireless sensor networks. These two protocol stacks are very similar and it is obvious that the wireless sensor network stack bears close resemblance to the more general stack. However, the wireless sensor network stack introduces some new concepts that the former stack is lacking. Below, we first present the more general protocol stack, which is the open system interconnection reference model (OSI) [20][1]; Secondly we present the wireless sensor network protocol stack [9].

3.4.1 Open system interconnection model

The OSI model provides a hierarchical structure to the address centric communication paradigm that is central in the design of data networks.

The hierarchical structure provides modularity, and this means that each hierarchical layer is a module with its own functions. Each module is conceptually viewed as a black box that provides service to the higher layer in the hierarchical structure. This makes it easy to replace a module with new or better functionality.

It is also clearly defined what a module at a particular layer is expected to do. The layers in this communication model are: Physical layer, Data link control layer, Network layer, Transport layer, Session layer, Presentation layer and Application layer. Figure 3.4 illustrates the layers in the OSI model and the data units used at different layers. The services or functionality each layer provides are presented with the lowest layer first and then succeeding higher layers.

**Physical layer:** The function provided by the physical layer is a virtual transmission link for transmitting a sequence of bits between any pair of nodes in the network. To be able to achieve this, a function that converts the representation of digital bits into a signal that can be transmitted over the communication channel is needed. It also needs to be a reverse function that converts the transmitted signal into a digital bit representation. The physical interface that performs both

\[
\mu(d) = \frac{N \pi d^2}{A},
\]

where \( N \) is the number of nodes distributed in a sensor region with area \( A \) and \( d \) can either be the range of a particular sensor or the radio transmission range (idealised with circular propagation).
of these functions are called a modem (This is from the words modulator and demodulator). The management of the virtual transmission link includes the establishment and termination of connections to the communication medium and participation in the process of efficiently sharing resources between users.

**Data link control layer:** The data link control layer provide an error free communication link between any pair of nodes in a network, i.e. point to point communication. That means that the physical layer provides an unreliable virtual transmission link that is converted at the data link control layer into a reliable virtual packet link for the higher layers. In essence the data link control layer works as follows: upon reception of a packet from a higher layer it adds a string of control bits to the beginning of the packet called a header and a string of bits is also added to the end of the packet called a trailer; this results in a string of bits that is longer than the initial packet and is called a frame. These added bits represent overhead in the transmission to correct errors that occurs in the physical layer, to request a retransmission when errors have occurred, to determine where the frame begins and ends, etc.

For the data link control layer to be able to handle multi-access communication a sublayer called the MAC sublayer is added to the bottom of the data link control layer. This sublayer manages the medium access control (MAC) for different nodes in the network. This is done so that transmissions can occur without

---

<table>
<thead>
<tr>
<th>Open System Interconnection model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host Layers</strong></td>
</tr>
<tr>
<td>Data</td>
</tr>
<tr>
<td>Application</td>
</tr>
<tr>
<td>Presentation</td>
</tr>
<tr>
<td>Session</td>
</tr>
<tr>
<td>Transport</td>
</tr>
<tr>
<td><strong>Media Layers</strong></td>
</tr>
<tr>
<td>Packet</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>Frame</td>
</tr>
<tr>
<td>Data Link Control</td>
</tr>
<tr>
<td>Bits</td>
</tr>
<tr>
<td>Physical</td>
</tr>
</tbody>
</table>

**Figure 3.4.** The open system interconnection reference model.
constantly interfering with each other.

**Network layer:** The major functions that the network layer provides are routing and flow control of packets in a network. The former is basically how the network layer decides, based on its network knowledge, the next node to which a packet needs to be transmitted in order to reach its destination in the network. The latter is fundamentally how to avoid sending data faster than the receiving node can absorb it, which in turn leads to congestion in the network i.e. an over usage of resources on that path in the network. Upon reception of a data packet from a higher layer it adds a packet header to that data packet. This header contains information such as source address, destination address, length of data packet, etc.

**Transport layer:** The transport layer has a number of functions, however, not all of these functions are needed in every network. The functions of the transport layer are as follows: It breaks messages into packets at the source node and at the destination node it reassembles the received packets into messages; multiple low-rate sessions at a source node that are destined to the same destination node can be multiplexed together into one session to the network layer; a high-rate session can be split into several lower-rate sessions to the network layer; it might be required for the transport layer to provide reliable end-to-end communication for sessions requiring it, if the network layer is unreliable; and lastly end-to-end flow control is usually handled at the transport layer. The transport layer usually requires a header appended to each packet and this so called data packet is passed to the network layer.

**Session layer:** The functions provided by the session layer deals with access rights in setting up sessions and to provide information about services to the transport layer that a users wants to access.

**Presentation layer:** The presentation layer provides the following major functions, which are: data encryption; data compression; and code conversion. The foremost function provide data encryption and it becomes exceedingly important in today’s networks as people, for example, transfer money over the network, accesses different servers, etc. The second function is that of data compression and it reduces the number of bits communicated in the network. As such more bandwidth is available for other sessions. The later is code conversion, which some times can be necessary due to incompatible character sets in, for example, different countries, terminals, etc.

**Application layer:** The last layer in the open system interconnection model is the application layer and it handles what is left after the other layers have performed their intended functions. Its major function is to perform tasks that are specific to a particular application.
3.4.2 Sensor network protocol stack

The open system interconnection model forms the basis for a wireless sensor network protocol stack. However, some of the functionality of the higher layers (i.e. session and presentation layer) are not needed for many applications and as such can be implemented if need arises at the application layer. Thus, the wireless sensor network protocol stack consists of five layers instead of seven that is used in the OSI-model. The five layers are: Physical, Data link control, Network, Transport and Application. The functionality of these layers has been discussed above.

In addition to this top-down five layer structure, there exists so called management planes, see Figure 3.5. These planes are connected to all of the available layers through well defined interfaces from and to which information can be both acquired from and allocated to. The information available at the management planes can then be used by all layers in the structure. This adds another dimension to the hierarchical structure normally used and it allows protocol and algorithm designers to utilise this extra knowledge. That is provided by these management planes to further optimise their algorithms. This extra knowledge can also be used at runtime to adapt algorithms to different circumstances and application needs. One of the most important aspects of the management planes is that the modularity of the layered structure is not broken, but is further extended. The available management planes in the sensor network protocol stack are the power, the task and the mobility management plane. These management planes are explained below.

Power management plane: This management plane handles how a sensor node manages its remaining power. An example is how to decide which components need to be active and which can be turned off at a given time based on the available knowledge. The knowledge obtained from the protocol stack helps scheduling these decisions, mainly for the transceiver. This can be by turning of the
transceiver to avoid getting duplicate messages, informing neighbouring nodes that message relaying can be accomplished using this node or not, etc.

**Task management plane:** The purpose of the task management plane is to balance and schedule the sensing tasks given to a region with some constraint, for example a life-time constraint. This can be accomplished since the number of sensor nodes is expected to be very large and thus all sensor nodes do not need to perform their sensing duties at all time. Some can be hibernating or even dormant until a predefined time or an activation signal is received. This scheduling and balancing may lead to that some nodes are active more than other nodes. This depends on their placement, the constraints assigned to that region, etc.

**Mobility management plane:** The mobility management plane registers movement and position of neighbouring nodes and detects and removes neighbouring nodes. This is done so a route can be obtained back to the end user. This knowledge is also used by the task management plane for power-, constraint- and task-balancing operations between nearby sensor nodes.

### 3.5 Medium access control

The first layer above the physical layer is the MAC layer and as such is influenced by many of its properties. The MAC layer manages the access to the shared medium such that packet transmissions from nodes do not cause to much interference with each other. Different kinds of MAC-layers have both advantages and disadvantages and the choice of MAC-layer depends on application specific requirements. Some of the traditional performance metrics are throughput, fairness, stability and delay. For wireless sensor networks a very important issue is energy conservation.

#### 3.5.1 Wireless issues

There are some unique properties of the wireless medium compared with the wired medium that makes it difficult and different to design MAC protocols. For sensor networks this will become even more difficult since nodes are energy constrained and to save precious energy a sensor node turns both on and off their transmitter on regular basis. These need to be taken into account during the design process of the MAC layer. Some wireless issues are presented below:

**Half-duplex operation:** In a wireless system the so called self-interference due to energy leakage into the received path makes it hard for the transceiver to both transmit and receive data at the same time. The energy leakage is usually much larger than the received signal strength, which then can not be detected. Due to the half duplex operation the uplink and downlink channels needs to be multiplexed in time (TDD) [58].
CHAPTER 3. SENSOR NETWORK ARCHITECTURE

Time varying channel: Radio signals propagate according to a phenomenon called multipath propagation. This phenomenon includes three distinct mechanisms: reflection, diffraction and scattering. Thus, the received signal is a superposition of time-shifts and attenuated versions of the transmitted signal and result in the variation of the received signal power over time. The rate of variation for the channel is determined by the coherence time. When the received signal power is below a predefined threshold the node is said to be in fade. During a fade the probability of errors is substantially increased and this results in the occurrence of long error bursts.

Location-dependant carrier sensing: Due to signal attenuation and multipath propagation, only nodes within a specific area of the transmitting node can detect a transmission. This results in three different node cases, explained below. The nodes must utilise a MAC layer that uses the carrier sensing technique for these problems to occur.

- A hidden node is a node that is within the range of the destination node, but is out of range of the sender. This is illustrated in Figure 3.6 a). Take an example, node A transmits a message to node B. However, when node C senses the channel it does not hear that node A is transmitting and falsely thinks that the wireless medium is idle. If node C initiates a transmission, it will interfere with the transmission from node A to node B. In this case, node C is a hidden node to node A and will for example cause unnecessary collision and lower the throughput of the communication medium.

- An exposed node is a node that is within the range of the sender, but is out of range of the destination node. This is illustrated in Figure 3.6 b). Consider an example in which node A transmits a message to node B. However, when node C senses the channel it hears the transmission from node A to node B and falsely thinks that the channel is busy. But any transmission initiated from node C does not reach node B. Theoretically node C can transmit messages to another node that is out of range of A but in range of C. Hence, node C is an exposed node to node A.

- Capture is when a receiving node can distinguish one of two or more simultaneous transmissions from different sources and successfully comprehend the transmitted message of that source. Using the capture technique can improve a protocols performance, but leads to unfair sharing of bandwidth with a preference for the nodes closer to the receiver.

3.5.2 Requirements

The traditional requirements of the MAC protocols have increased and new requirements are being added as the usage of the networks evolves and the networks them-self
have changed. From the more traditional requirements as, for example, throughput and delay which still are important, to a requirement such as energy efficiency, which is very important in wireless sensor networks. Thus, below follows a description of the traditional requirements first. Then the additional requirements for wireless sensor networks are described and it also highlights how the traditional requirements importances has changed in these types of networks.

**Traditional**

Traditional requirements of MAC protocols focuses on a number of performance metrics which make it possible to do good comparisons between the different protocols. Some of the more important metrics are described below and these are throughput, delay, stability and fairness. Additional metrics can be robustness to fading, power consumption and quality of service (QoS) support. However, the additional metrics was initially not used for comparisons between the protocols.

**Throughput:** The fraction of the channel capacity used for data transmission is defined as the throughput. This means that it is only the relevant information that is taken into account and that the protocol overhead is not considered when calculating the throughput. The goal for a MAC protocol is to maximise the information throughput and that also means to minimise the access delay to the channel.

**Delay:** The average time spent in the MAC queue for a frame before transmission. This is referred to as the access delay for that MAC protocol i.e. the time instance from it is en-queued until it is successfully transmitted and thus de-queued. One important thing to remember when comparing MAC protocols with respect to delay is that the delay is a function of network and traffic characteristics.

**Stability:** Due to variations in the traffic and network characteristics a stable system should be able to handle instantaneous load capacities above the maximum

---

**Figure 3.6.** a) the hidden node problem and b) the exposed node problem.
transmission capacity of the channel. When the systems average load is below the transmission capacity of the channel.

**Fairness:** Fairness is defined for non prioritised traffic as when multiple nodes try to access the channel they have an equal probability of gaining its access i.e. no node that is trying to access the channel is prioritised. This results in fair sharing of the available bandwidth.

**Wireless sensor network**

In wireless sensor networks some of the traditional requirements like fairness, throughput and delay are shadowed by the need to conserve energy for a sensor node. The reason for this is that major energy savings can be obtained by turning of the transceiver in a battery operated sensor node. The traditional requirement of fairness in a sensor network is mostly not an issue due to the collaboration towards a common goal. However, if there is more than one goal in a sensor network this could be an increasing problem. As for the delay, it is traded against increased energy savings. Throughput is mostly not an issue today, but will be increasingly more important due to quality of service guaranties in future sensor networks.

In addition to conserve energy other additional requirements in wireless sensor networks are scalability and robustness to frequent topology changes. The former is important due to the dense deployment of sensor nodes with hundreds or even thousands of nodes in a sensor field. The latter can be caused by the need to conserve energy i.e. powering down the transceiver, death of existing nodes and deployment of new nodes. Some of the issues related to energy conservation at the MAC protocol for wireless sensor networks are unnecessary collisions, overhearing, idle listening and protocol overhead [34]. These issues are described in more detail below.

**Overhearing:** When a source node transmits a frame to one destination node, called unicast, over the wireless medium all nodes within the transmission range are able to receive the packet. However, the packet is dropped if it is not addressed to that node. It leads to unnecessary packet reception, called overhearing. In dense networks overhearing a transmission can result in major energy waste and thus avoiding this issue is important. In some cases overhearing can be desirable; an example is to estimate the current traffic load in the network for management purposes.

**Collisions:** Reducing the number of collisions in a network can save considerable amounts of energy. This is due to the energy consumed at the receiving and the transmitting nodes, which is consumed in vain because a retransmission might be need for that frame. The problem of collisions can be solved or reduced by the use of appropriate algorithms. The solution includes for example fixed assignment, demand assignment and collision avoidance protocols. These protocols are explained in more detail in the next subsection.
3.5. MEDIUM ACCESS CONTROL

Figure 3.7. MAC-protocol classification.

Protocol overhead: Protocol overhead for the MAC layer includes all types of infrastructure communication that originates from this layer. This includes for example acknowledgement, clear to send (CTS) and request to send (RTS) frames. It also includes packet header and trailer of the data frames.

Idle listening: When the transceiver is ready to transmit or receive frames, the transceiver is said to be in the idle state. This idle state consumes much power, not as much as when transmitting or receiving frames. But, it is larger than turning the transceiver off or into a sleep state. Thus, much power can be saved by avoiding the idle state. One example that can help minimising the idle listening is a concept called wake-up transceivers, which is explained in Chapter 2.1.3.

The MAC protocols developed so far for wireless sensor networks have used some or all of the concepts related to energy savings.

3.5.3 Categories of MAC protocols

MAC protocols can be classified [25] into two broad classes namely: centralised protocols, and distributed protocols. This classification is based on the need for a central coordinator (i.e. a sink) or not. These protocols can further be categorised into guaranteed, random and hybrid access protocols. This categorisation is illustrated in Figure 3.7 and is explained in more detail below.

Guaranteed access protocols

Guaranteed access protocols divide the available resources between all possible communications nodes. This is done in such a way that each node has an exclusive share of resources under its disposal. This assignment of available resources can be both on
a long-term or short-term basis. The former is in the order of minutes, hours or even
longer periods of time and the latter is usually just for a series of data transmissions.

Protocols that are based on long-term resource allocation are for example time
division multiple access (TDMA), frequency division multiple access (FDMA) and
code division multiple access (CDMA). These protocols [58][67] divide the available
resources in different ways. The TDMA scheme uses a fixed length superframe that
spans a predefined time interval. After this time interval has elapsed the superframe
is repeated again. These super frames are subdivided into a number of fixed length
frames (i.e. time slots) that are assigned exclusively to the available nodes. The man-
agement of the mentioned scheme requires time synchronisation between nodes to
avoid overlapping timeslots, which results in interference between nodes. In contrast,
the FDMA scheme divides the available resources into different frequency bands i.e.
the available frequency band is subdivided into subchannels, which are assigned ex-
clusively to the available nodes. This scheme requires a more complex transceiver and
frequency synchronisation. At last, the CDMA scheme uses orthogonal codes for each
node to spread the signal over a much larger bandwidth. These codes separate the
transmission for each node from the other nodes and as such the receiver must know
the code used by the transmitter to be able to decode the message successfully. All
other nodes appear as noise to the receiving node. However, the code management is
an important aspect when using this scheme.

Protocols based on short-term resource allocation can either use a master-slave
configuration or by exchanging tokens with each other. The former uses a master node
that polls the slave nodes. The slave node then transmits its data in response to that poll.
In these protocols the master node must be active during the entire polling sequence
and as such consumes a lot of power relative to the slave nodes. These protocols are
referred to as polling protocols [37][46]. The latter exchanges a token and only the
node with the token is allowed to transmit data. After the node has transmitted its data;
it passes the token to the next node in line, which is then allowed to transmit its data.
Due to the variation in circulation time of the token, the transceivers must be active at
all time or its operation carefully scheduled. The losses of a token in these types of
protocols, due to channel errors, causes a lot of overhead. This relates to the recovery
of the token. These protocols are called token-passing protocols [18].

**Random access protocols**

In this type of protocols the nodes contend for the wireless medium and as such they
are uncoordinated and can operate in a fully distributed manner. If only one node
tries to access the channel it results in a successful transmission of the frame to the
destination node. However, if two or more nodes try to access the channel at the same
time a collision occurs and none of these nodes successfully delivers their frames to the
intended destination nodes. To prevent these nodes from trying to access the channel
at the same time again, a collision resolution algorithm (CRA) [30][10] is used. This
algorithm resolves the collision in an orderley manner, according to the rules defined
3.5. MEDIUM ACCESS CONTROL

by the CRA. This is usually accomplished by the use of a random backoff time, due to
the nature of the wireless medium.

One of the first random access protocols is the ALOHA protocol developed at the
University of Hawaii for packet radio networks. It is a classic example for this type of
protocol and it operates as follows: If a node has a frame to transmit, it transmits the
frame directly. If no other node is transmitting a frame at the same time, it successfully
delivers the frame to the destination node. Otherwise, a collision occurs and each of
these nodes waits a random period of time before they try to transmit their frame again.
There is an improvement called slotted ALOHA, in which the time is divided into slots,
so called timeslots. A transmission may only start at the beginning of a timeslot and as
such the collision interval is halved. This roughly doubles the throughput of the slotted
ALOHA system compared to that of ALOHA.

Further improvements have been suggested to the ALOHA protocol [15], one of
which is that nodes wanting to transmit a frame is respectful of ongoing transmissions.
Basically, this means that a node is required to senses the channel before it initiates
a transmission; this is called carrier sensing. If it detects an ongoing transmission it
defers from conducting a transmission before it tries again according to the CRA used;
otherwise it initiates a transmission. This is the procedure used in the carrier sense
multiple access (CSMA) protocol [15], which is commonly used in today’s wired and
wireless networks.

One of the problems in wireless networks, as mentioned previously, is location-
dependent carrier sensing or more precisely the problems that occur with hidden and
exposed nodes. To combat these problems collision avoidance (CA) algorithms have
been proposed and some of the available solutions include control handshaking and
out of band signalling. These two important techniques are explained in more detail
below.

Control handshaking: This technique is used in for example the multi-access col-
lision avoidance (MACA) protocol [60] and in the IEEE 802.11 protocol [2].
The former uses a three way handshaking mechanism, in which the source node
transmits a request to send (RTS) frame, which is received by all nodes in the
transmission area of the source node; the destination node responds with a clear
to send (CTS) frame that is received by all nodes within the transmission area of
the destination node; upon reception of a CTS frame the source node assumes
that the channel is acquired and initiates its data transmission. Nodes that receive
either a RTS or a CTS frame defers from doing a transmission. This procedure
eliminates the hidden node problem to a large extent. Another problem with
this technique is the number of control frames needed to be transmitted incurs a
significant overhead, especially in case of short data frames.

Out of band signalling: These techniques use two channels, which are divided in fre-
quency [65]. One channel is the data channel and the other one is used for control
signalling. They basically operate as follows. Before a node starts a transmis-
sion it listens to the control channel to verify that no other node is transmitting
a frame. If the control channel is idle it initiates the data transmission; otherwise it uses some CRA before it listens to the control channel again. Upon receiving a data frame, the receiving node emits a busy signal on the control channel and ends this signalling once the frame is completely received. This protocol minimises the number of exposed nodes. However, this technique does not completely solve the hidden node problem. A technique in which both the transmitting node and receiving node transmit a busy signal on the control channel eliminates the hidden node problem, but increases the number of exposed nodes. Problems with this technique include the use of more than one channel.

**Hybrid access protocols**

The hybrid access protocols try to use the best from both the random access and the guaranteed access protocols. These protocols are usually based on the request-grant mechanism that works as follows. A node sends requests of a resource allocation to a central coordinator using a random access protocol. The central coordinator then allocates that resource reservation to an upstream channel and sends a grant to that node. This grant indicates the reserved resource allocation, which is exclusive to that node.

These hybrid access protocols can further be divided into random reservation access (RRA) and demand assignment (DA) protocols. This is based on the intelligence of the central coordinator. In the former protocol, the central coordinator has distinct rules of how the resource allocation is conducted. This can for example be a successful request results in one resource constrained allocation with a time limit. The latter protocol collects all requests from nearby nodes and schedules these requests in an orderly manner. This can for example be based on QoS requirement of the data.

An example of a hybrid protocol that uses a central coordinator is the IEEE-802.15.4 protocol standard [3][26] and specifically when it operates in beacon enabled mode. This protocol divides the time between each beacon into an inactive and an active part. The former allows nodes to turn off their transceivers to save energy. The latter consists of a superframe that is divided into 16 timeslots. These timeslots are divided into a collision access period and a collision free period. In the collision access period both control and data frames are transmitted using a slotted random access protocol. In this period, requests to the central coordinator for one or more dedicated time slots in the guaranteed access period are conducted. Nodes that have successfully been granted one or more time slot in the collision free period send their data at those time slots. The beacon is transmitted by the central coordinator at the first available time slot and one of its uses is synchronisation. This protocol is mainly used for energy constrained sensor nodes that are connected to an energy rich central coordinator.
3.6 ZigBee

ZigBee [23][22] is developed and enhanced by the ZigBee Alliance and is designed for wireless personal area networks (WPAN). The ZigBee protocol stack uses the IEEE 802.15.4 standard, which defines the physical and the media access control layers, see Figure 3.8. At the network layer ZigBee uses protocols that can utilise full mesh and star topologies. The mesh topology allows nodes to connect and communicate directly with other nodes in range. Only the intended recipient is able to act upon the content of the transmitted messages. This topology further allows communication to pass through any number of nodes between source and destination nodes. The star topology on the other hand offers low-latency connections. ZigBee also provides a security layer that uses the 128 bit advanced encryption standard cryptography and trust-center-based authentication. The application framework provides a set of functionality that developers can utilise to construct application profiles.

3.6.1 IEEE 802.15.4

The IEEE 802.15.4 standard [26] defines the physical and the MAC layers in the OSI-model. There are two physical layers defined (i.e. the 868/915 PHY and the 2.4 PHY) and both are based on direct sequence spread spectrum techniques. These techniques result in low-cost implementations of integrated circuits. The main differences between the two physical layers are the frequency bands and the data rates. The 868/915 PHY uses the 868 MHz frequency band with a data rate of 20 kbits in Europe; and the 915 MHz frequency band with a data rate of 40 kbits is used in the United States. The 2.4 PHY operates worldwide at the 2.4 GHz frequency band (ISM band) and uses a data rate of 250 kbits. The 2.4 GHz band offers advantages such as a larger market and as such a lower manufacturing cost. While the 868 MHz and 915 MHz bands offer
CHAPTER 3. SENSOR NETWORK ARCHITECTURE

an alternative to the congestion and interference (e.g. WiFi, Microwave ovens, etc.) of the 2.4 GHz band and a longer transmission range for a given link budget due to a lower propagation loss.

The MAC layer provides features such as association and disassociation, acknowledged frame delivery, frame validation, channel access mechanism, guaranteed time slot management and beacon management. The use of some of these features depends on if it is operating in the beacon enabled mode or not. Both the guaranteed time slot management and the beacon management are only used in the beacon enabled mode. However the channel access mechanism varies depending on if the beacon enabled mode is used or not. If the beacon enabled mode is used the channel access mechanism is slotted CSMA/CA; otherwise non-slotted CSMA/CA is used. Both of these mechanisms sense the channel before they transmit the message. If a collision occurs the collision avoidance procedure randomly delays the transmission before it senses the channel again.
Chapter 4

Energy Dissipation Models for Wireless Sensor Networks

In this chapter we describe dynamic power management and three energy dissipation models for wireless sensor networks. These energy models are: the radio energy dissipation model; the state based energy model; and the complete energy dissipation model. The former only uses the communication layer to derive its two operation states, namely the transmit and the receive states. While the two latter energy dissipation models use all layers in the dynamic power management model to derive their operation states. Following these descriptions are measurements on a wireless sensor node developed at the communication electronics group at Linköping University. These measurements include the power consumption for the operation states and the state transitions as well as the time to complete a transition. Furthermore, the complete energy model is evaluated using these measurements. Then a case study is conducted using the complete energy dissipation model.

4.1 Dynamic power management

The basic idea with dynamic power management (DPM) is to deactivate components when they are not needed and to activate them again when they are needed. The difficult part is to schedule the activation and deactivation of components in such a manner that they are active exactly when they are needed and inactive in the remaining time. This is difficult because it is hard to predict when something is going to happen in the future and components do not start instantaneously but require an activation time, which depends on the components used.

From the basic sensor node architecture there are three subsystems that are possible to put into different types of low power states. These subsystems are the sensing, the communication and the processing subsystem. Two of these subsystems can be considered primary subsystems namely the sensing and the communication subsystems. This is due to the fact that these subsystems provide added capabilities to the sensor node
like communication and sensing, while the processing subsystem provides services in the form of information processing, storage capabilities to the primary subsystems. The processing subsystem also controls the other subsystems using the appropriate drivers. Thus it is preferable that the processing subsystem operates in conjunction with other primary subsystems in the sensor node and not by itself. In this case, a sensor node should be able to sense its surrounding, communicate wireless with other sensor nodes and enters some kind of low power state to prolong its lifetime.

From this we can deduce three layered states to our DPM model for wireless sensor nodes which are: sleep, sense and communication states. The sleep state is the most energy efficient state since both primary systems are turned off and the processing subsystem is assigned to its sleep mode. It serves the purpose of prolonging the sensor nodes lifetime. The main purpose of the sense state is for the sensor node to sense its surrounding without distributing its collected data to other nodes in the network. This provides the ability to aggregate data locally over a period of time. In this state the communication subsystem is turned off, but both the sensor and the processing subsystems activity depend on the number of sensors available and how they are managed by the processing subsystem. This depends on both hardware and software design choices. The communication state consumes the most power of all these states. It extends the functionality of the sense state by providing the ability to distribute the collected data and to relay data collected by other nodes to their intended destination. This state the communication subsystem is active, the activity of the sensor subsystem depends on the application requirements and the processing subsystems activity depends on the requirements of both the communication and the sensor subsystem.

These three states form a three layer DPM model where each higher layer provides added capabilities and increased power consumption compared to the lower layers. The power consumed in these layered states depends on the actual hardware components and the behaviour of the software that controls it. These major states may need to include sub-states to accommodate the wide variety of states that these subsystems can enter. If we take the communication subsystem as an example, the hardware states usually provided by the manufacturer of an active radio chip are transmit, receive and idle states. There can however be an additional number of low power states that increase the activation time of the radio chip. How these states are used depends mainly on the application requirements for the wireless sensor network and the individual nodes.

### 4.2 Radio energy dissipation model

The energy dissipation model described in [27] is a radio energy dissipation model that only takes into account the transmit and receive circuitry and the transmit amplifier in the transceiver. Thus, this model only takes into account the actual transmission and reception of data. The reason to only include an active transceiver in this energy dissipation model is due to the fact that a transceiver consumes the majority of the energy in a sensor node. Another fact is that communication protocols are evaluated
4.2. RADIO ENERGY DISSIPATION MODEL

with respect to their communication characteristics. This includes both application and infrastructure communication. It is important to notice that different assumptions about the radio characteristics changes the advantages and disadvantages of different protocols. This radio energy dissipation model can be used to compare the energy consumed for communication in, for example, different routing, cluster, and location estimation protocols.

To view this in another perspective, let’s consider the DPM model described in 4.1. By using this energy model the only important layer is the third one and as mentioned before it is also the most demanding layer with respect to energy consumption and processing power. As such, this model only takes into account the communication subsystem and neglects the other subsystems.

4.2.1 Energy model

This radio energy dissipation model takes into account the energy consumed in the radio circuitry and the transmit amplifier, as mentioned above. Since the energy consumed during a transmission in the transmit amplifier depends on the distance. A propagation model is used to model the signal power attenuation over that distance. This energy model uses the free space propagation model [58] when the transmission distance $d$ is less than the cross-over distance $d_{co}$ and the ground reflection (2-ray) propagation model [58] when $d$ is larger than $d_{co}$.

The Friss free space equation is used when $d < d_{co}$ and is

\[ P_{Rx}(d) = P_{Tx}G_{Tx}G_{Rx} \frac{\lambda^2}{(4\pi d)^2 L}, \tag{4.1} \]

where $P_{Rx}(d)$ is the received power at a transmitter-receiver separation distance of $d$, $P_{Tx}$ is the transmit power, $G_{Tx}$ is the transmitter antenna gain, $G_{Rx}$ is the receiver antenna gain, $\lambda$ is the wavelength of the carrier signal and $L \geq 1$ is the system loss factor.

The two-ray ground reflection equation is used when $d \geq d_{co}$ and it is

\[ P_{Rx}(d) = P_{Tx}G_{Tx}G_{Rx} \frac{h_{Tx}^2 h_{Rx}^2}{d^4}, \tag{4.2} \]

where $h_{Tx}$ is the height of the transmitter antenna and $h_{Rx}$ is the height of the receiver antenna.

From equations (4.1) and (4.2) the crossover distance is obtained when the received power from both of these propagation models are equal. Thus, it is defined as:

\[ d_{co} = \frac{4\pi \sqrt{Lh_{Tx}h_{Rx}}}{\lambda} \tag{4.3} \]
The block diagram of the radio energy dissipation model is shown in Figure 4.1. In this model to transmit a $m$-bit message over a distance $d$, the transmitter consumes

$$E_{Tx}(m, d) = E_{Tx,el}(m) + E_{amp}(m, d)$$

$$= \begin{cases} 
  m(E_{el} + \epsilon_{f-amp}d^2) & \text{if } d < d_{co} \\
  m(E_{el} + \epsilon_{2-amp}d^4) & \text{if } d \geq d_{co}
\end{cases}$$

(4.4)

and to receive that $m$-bit message, the receiver consumes

$$E_{Rx}(m) = E_{Rx,el}(m)$$

$$= mE_{el}$$

(4.5)

where $E_{Tx} = E_{Rx} = E_{el}$ is the energy consumed to run the transmitter or receiver circuitry, $\epsilon_{f-amp}$ is the energy consumed by the transmit amplifier when $d < d_{co}$ and $\epsilon_{2-amp}$ is the energy consumed by the transmit amplifier when $d \geq d_{co}$.

The parameters used for this energy model are presented here according to the experiments in [27], which are $E_{el} = 50\, \text{nJ/bit}$, $\epsilon_{f-amp} = 10\, \text{pJ/bit/m}^2$ and $\epsilon_{2-amp} = 0.0013\, \text{pJ/bit/m}^4$.

The parameter $E_{el}$ is the energy consumed per bit for both the transmitter and the receiver circuitry and depends on factors such as digital coding, modulation and filtering. At that time, researchers had designed transceivers that supported direct-sequence spread spectrum communication and one such transceiver expanded $165\, \text{mW}$ in transmit mode and $46.5\, \text{mW}$ in receive mode. As such, they assumed that a reasonable energy consumption for a 1 Mbps transceiver is $50\, \text{mW}$ when operating in both transmit and receive modes.

The two parameters $\epsilon_{f-amp}$ and $\epsilon_{2-amp}$ depends on the receiver sensitivity and the receiver noise figure to be able to adjust the transmit power, such that the received power is over a reception threshold at the receiving node. This is accomplished by working from the received power threshold, $P_{Rx-th}$, to determine the minimum transmit power. Since we know that the transmit power:

$$P_{Tx} = E_{amp}(m, d)R_b$$

(4.6)

where $R_b$ is the communication bitrate. By substituting the received power $P_{Rx}$ with $P_{Rx-th}$ and the transmit power $P_{Tx}$ with that in equation (4.6) into equations (4.1)
and (4.2) depending on the propagation model used. From this we can determine the value for the parameters $\epsilon_{f-amp}$ and $\epsilon_{2-amp}$ and results in the two following equations:

$$\epsilon_{f-amp} = \frac{P_{Rx,th}(4\pi)^2}{R_b G_t G_r \lambda^2}$$  \hspace{1cm} (4.7)$$

$$\epsilon_{2-amp} = \frac{P_{Rx,th}}{R_b G_t G_r (h_t h_r)^2}$$  \hspace{1cm} (4.8)$$

From equations (4.7) and (4.8) the consumed power per bit relative to a transmission distance can be calculated. The values used for calculating this are: $P_{r,th} = -52 \text{ dBm}$, $R_b = 1 \text{ Mbps}$, $h_t = h_r = 1.5 \text{ m}$, the antenna gain at the transmitter $G_t = 1$, and the antenna gain at the receiver $G_r = 1$. Using these values results in the presented energy consumption values for the transmit amplifier above.

### 4.2.2 Scenario

In this scenario the radio energy dissipation model is used to compare direct communication with multihop communication using a simple linear network model shown in Figure 4.2. This network model uses $n$ nodes and these nodes are equally separated with a distance $d$, in meters. The data originates from node $n$ i.e. the node furthest away from the sink. The maximum transmission distance between node $n$ and the sink is $d_m = nd$ meters and it is the transmission range used for direct communication. The multihop communication, on the other hand, uses the nodes that are in between the sink node and node $n$ (i.e. node $i$, where $1 < i < n$) to relay its data and as such each hop has a range of $d$ meters. This comparison uses the previously presented parameters for this energy model with a calculated crossover distance of $86.2 \text{ m}$.

For direct communication the energy consumed for the radio varies depending on the distance from node $n$ to the sink node, according to equation (4.9).

$$E_{direct} = E_{tx}(m, d_m) + E_{rx}(m)$$

$$= \begin{cases} m(2E_{el} + \epsilon_{f-amp}d_m^2) & \text{if } d_m < d_{co} \\ m(2E_{el} + \epsilon_{2-amp}d_m^4) & \text{if } d_m \geq d_{co} \end{cases}$$  \hspace{1cm} (4.9)$$
CHAPTER 4. ENERGY DISSIPATION MODELS

Figure 4.3. Comparing direct versus multihop communication.

On the other hand, in the multihop communication approach, aside from depending on the distance, it also depends on the number of hops taken. Thus, the multihop communication approach consumes energy according to the following equation:

\[ E_{\text{multihop}} = nE_{\text{Tx}}(m, d) + nE_{\text{Rx}}(m) \]

\[ = \begin{cases} 
    nm(2E_{\text{el}} + \epsilon_{f-\text{amp}}d^2) & \text{if } d < d_{co} \\
    nm(2E_{\text{el}} + \epsilon_{2-\text{amp}}d^4) & \text{if } d \geq d_{co}
\end{cases} \quad (4.10) \]

Comparing direct communication with multihop communication is straightforward and this is visualised in Figure 4.3. From this figure, it can be seen that direct communication (i.e. when 1 node is used) is preferable when the transmitting node is within 105 meters from the sink node. Otherwise, multihop communication is preferable. It is important to remember that this result is very dependent on the parameters chosen for this energy model and other parameters may give different results in the matter of choosing the communication approach (i.e. direct or multihop communication).

Furthermore, an expression is given for when multihop communication consumes more energy than direct communication i.e. \( E_{\text{direct}} < E_{\text{multihop}} \). This is easily obtained from equations (4.9) and (4.10). The result is shown in equation (4.11) below. This
4.3 State-based energy dissipation model

The state-based energy dissipation model (SEDM) [44] takes into account the energy dissipation for the sensor and the communication subsystem in a sensor node. As such, it does not specify the activity of the processing subsystem. The processing subsystem usually consumes much more energy than that of the sensor subsystem, which makes it difficult to motivate why it has been neglected. This energy model can be used to analyse the performance of, for example, different energy map construction algorithms and sleep scheduling policies.

This model uses all three layers of the DPM model presented in section 4.1, but as mentioned above it neglects the processing subsystem activity. It also assumes that all transitions between the different states are instantaneous even though events can occur during a transition. This energy model is described in more detail below.

The SEDM is represented by the use of four operation states shown in Table 4.1. The four states describe the activity of the sensor and the communication subsystems in a sensor node, but not the activity of the processing subsystem. The processing subsystem is one of the more demanding subsystems in addition to the communication subsystem. The different states are explained below:

1 Sleep: In this state the sensor and the communication subsystem is deactivated (turned off). It is also the deepest sleep state in the SEDM.

2 Sense: The sensor subsystem is active (turned on) and the communication subsystem is deactivated.

3 Receive: The communication subsystem is set into receive mode and the sensor subsystem is active.

4 Transmit: The sensor subsystem is active (turned on) and the communication subsystem is set into transmit mode.

Table 4.1. The four operation states used in SEDM.

<table>
<thead>
<tr>
<th>Operation state</th>
<th>Sensor</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Sleep</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>2: Sense</td>
<td>On</td>
<td>Off</td>
</tr>
<tr>
<td>3: Receive</td>
<td>On</td>
<td>Rx</td>
</tr>
<tr>
<td>4: Transmit</td>
<td>On</td>
<td>Tx</td>
</tr>
</tbody>
</table>

assumes that $d_m < d_{co}$, which is not the case in Figure 4.3.

$$E_{direct} < E_{multihop}$$

$$2E_{el} + \epsilon_f \times d_m^2 < 2nE_{el} + n\epsilon_f \times d^2$$

$$d^2(n^2 - n)\epsilon_f \times \frac{d^2}{2(n - 1)} < E_{el}$$

(4.11)
4 Transmit: This state has an active communication subsystem that is set into transmit mode and the sensor subsystem is active. This is also the state that consumes most power.

The state transition diagram shown in Figure 4.4 describes the transition between the available states. However, it is assumed that the state transitions are instantaneous. In addition to the four operation states used, two additional states $2'$ and $3'$ are also shown in the transition diagram. The difference between these two states and the states $2$ and $3$ are that no timer is started when entered. Instead it verifies if there are any events present that need to be executed. In terms of power consumption both state $i$ and $i'$ are the same, the difference are the behaviours of these states in the transition diagram.

The diagram of Figure 4.4 shows the event path of a state transition for a sensor node utilising this model. Thus, each transition constitutes of a number of commands that needs to be tested before a new state is reached. The available test are: a) routing, evaluates the need for a message to be routed; b) event exists, determines if there are any sensor events present; c) turn radio on, evaluates the need for the sensor node to turn on the radio; d) receive, determines if the communication subsystem needs to transmit or receive data; and e) sleep, decides if the sensor node can enter the sleep state to conserve energy. Timer is an action that initiates a delay with a predefined time. Each of the tests performed during a state transition, are evaluated based on a probabilistic parameter that tries to capture the behaviour of a sensor node.
Table 4.2. The four operation states used in this energy model for wireless sensor networks. 

<table>
<thead>
<tr>
<th>Operation State</th>
<th>Sensor</th>
<th>Processing</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Sleep</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>2: Sense</td>
<td>Active</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>3: Transmit</td>
<td>Active</td>
<td>Active</td>
<td>$T_x$</td>
</tr>
<tr>
<td>4: Receive</td>
<td>Active</td>
<td>Active</td>
<td>$R_x$</td>
</tr>
</tbody>
</table>

The outcome of a test is tied to the corresponding event. This means that the outcome of the test *routing* is yes, if the node needs to relay data from other nodes in the network and no otherwise. This also holds true for both of the tests *event* and *turn radio on*. The former is yes if an event is present and the latter is yes if the radio needs to be turned on. The *receive* event depends on its characteristics and the parameter is influenced by the degree of cooperation needed by the application. If the test of *event* is no, the *sleep* test is called, which reflects that no sensor events are present. This test depends on the degree of coverage needed by the application. The greater the *sleep* parameter is, the smaller the coverage will be.

### 4.4 Complete energy dissipation model

The complete energy dissipation model [49][48] takes into account all subsystems used in the basic sensor node architecture, which results in four states. It allows this energy model to model the sensor node characteristics more accurately. In addition, it also includes the time and the expended power to complete a state transition such that the scheduling between the different states can be accomplished without wasting energy. This energy model can be used, for example, to determine if it is viable to switch state based on the available knowledge, to determine the energy consumption for MAC, routing and location estimation protocols, etc.

This energy model also uses all layers in the DPM model described in section 4.1. The assumptions and the energy model are presented in the following section.

### 4.4.1 Energy model

From the three layered states in the DPM model we construct our complete energy model, by assuming that the sensor and processing subsystems can utilise two states, active and off. The communication subsystem has three possible states to use, namely off, transmit and receive. From the above assumption, the idle state in the communication subsystem is not used. This is because it is preferable, from an energy point of view, to scan the wireless channel for incoming transmissions, receive or transmit data than to be in the idle state in which the communication subsystem can be deactivated. Though the idle state needs to be entered to be able to either transmit or receive data
CHAPTER 4. ENERGY DISSIPATION MODELS

Figure 4.5. State transition diagram of the complete energy dissipation model.

and thus it is included in the appropriate state transitions. The proposed energy model has a three layer structure which is represented by the use of four different operation states shown in Table 4.2 and the state transition diagram is shown in Figure 4.5. The four operation states are: sleep, sense, receive, and transmit. There exist two directional transitions between each pair of operation states.

Operation states

In this subsection a description of the different operation states and their energy consumption are presented. The operation states used in this energy model are presented in order of appearance in Table 4.2.

The sleep state is used to prolong the life time of a sensor node when no activity such as communication and collection of data is needed. This can occur when other nodes in the vicinity collects data, which is known to be correlated to this particular sensor node or there is nothing to report and the relaying of data can be handled by other nodes in the vicinity. As such, all primary subsystems in the sensor node are deactivated and the processing subsystem enters a deep sleep state.

The purpose of the sense state is to collect data from its sensors and to store the data for further processing at a later time. Due to the limited processing requirements of the sensor subsystem, which mainly involves storage of local data, the processing subsystem energy consumption can be neglected. One reason for this is in case of event detection where the sensor subsystem activates the processing subsystem using an interrupt. After the interrupt is received the processing subsystem initiates a state transition to the transmit state so that the event can be reported directly. Another reason
is that some sensors today can store at least one reading in the sensor subsystem and from an energy perspective it would be good to extended this in the future to allow sensors to store all their readings in the sensor subsystem without using the processing subsystem as a relay. In this state the communication subsystem is deactivated; the sensor subsystem is active; and the processing subsystem is set to one of its low power modes.

The activity of the receive and transmit states is very similar and they mainly differ in the activity of the communication subsystem. In both of these states the sensor subsystem is active and collects data for further processing and the activity in the processing subsystem is execution of communication protocols, signal processing, data aggregation, etc. The receive state exists for the sensor node to be able to receive data or commands from other nodes in the network that are either destined to this node or are going to be relayed to another node in the network. The transmit state is used to transmit information to other nodes in the network either its own data or data received from other nodes. In this state the communication subsystem is set to transmit mode. It is important to note that the communication between nodes is half-duplex because of the added complexity, cost, and the increased energy consumption for full-duplex communication.

The energy consumption in each operation state depends on the activity of each subsystem and the time spent in that operation state. We now define the concept of a visit as the time duration between the arrival and the departure at one operation state. Since the time duration for each visit in operation state $i$ can vary, the total accumulated time $T_i$ in operation state $i$ is divided into a number of visits. Let $\tau_i(k)$ denote the time spent in operation state $i$ during the $k^{th}$ visit and thus $T_i = \sum_{\forall k} \tau_i(k)$. The energy consumed in operation state $i$ can be obtained as

$$E_i = P_i T_i = P_i \sum_{\forall k} \tau_i(k)$$  \hspace{1cm} (4.12)

were $P_i$ is the average power consumed at operation state $i$. The description of how to obtain $P_i$ for these operation states are presented below.

**Sleep state:** The sleep state can be seen as a state of hibernation where all subsystems are either turned off or are set into a deep sleep mode. Before entering the sleep state the processing subsystem needs to activate a timer that schedules an interrupt to be delivered to the processing subsystem at the time of reactivation. Since, there is minimal activity in the sensor node, which mainly is due to the activity of the timer and its related circuits. The average power expended in the sleep state is obtained as follows

$$P_1 = P_{\text{leakage}} + P_{\text{timer}}$$  \hspace{1cm} (4.13)

where $P_{\text{leakage}}$ is the average leakage power and $P_{\text{timer}}$ is the average power consumed to operate the timer circuits. The time duration in this state, $\tau_1(k)$,
is usually a predetermined parameter that states how long the sensor nodes is in hibernation before reactivation.

**Sense state:** The sense state is a low power state, which the sensor node enters when its tasks are related to data gathering and storage of gathered data. The active subsystem is the sensor subsystem. The processing subsystem is set into a low power mode since the minimal activity needed to store the data can be neglected. The communication subsystem is deactivated since no communication is necessary. The average power consumed by the sensor node in this operation state is

\[
P_2 = P_1 + P_{\text{sensor}}
\]  

(4.14)

where \(P_{\text{sensor}}\) is the average power consumed by the sensor subsystem. The time duration in this state, \(\tau_2(k)\) is a predetermined parameter that decides how long the sensor node should gather data about its surrounding. This parameter includes analog to digital conversion, storage, and event detection.

**Transmit state:** The transmit state is the operation mode that the sensor node enters when it is going to transmit data to other sensor nodes in the vicinity. The major power expended in this mode is mainly due to the transmitter electronics and the transmit amplifier in the communication subsystem. The transmit amplifier consumes power related to the distance between the transmitting node and the receiving node to achieve an acceptable signal to noise ratio. In addition to the activity in the communication subsystem, both the sensor and the processing subsystem are active. The power expended by the sensor node in this operation state is

\[
P_3 = P_2 + P_{pr} + P_{Tx} + P_{\text{amp}}
\]  

(4.15)

where \(P_{Tx}\) is the average power consumed by the transmitter electronics, \(P_{\text{amp}}\) is the average power consumed by the transmit amplifier. The time duration in this state, \(\tau_3(k)\), depends on the transmitting time and an additional signal processing time.

**Receive state:** The receive state is the operation mode that the sensor node enters when it needs to receive information from other nodes in the network and the major power expended in this state is due to the activity of the receiver electronics in the communication subsystem. Both the sensor and the processing subsystems are active in this state. The average power consumed in this state is

\[
P_4 = P_2 + P_{pr} + P_{Rx}
\]  

(4.16)

where the \(P_{pr}\) is the average power consumed for an active processing subsystem and \(P_{Rx}\) is the average power expended by the receiver electronics. The time duration for the sensor node in this state, \(\tau_4(k)\), is a parameter that depends on the listening time, the receiver waiting time, the receiving time and additional time for signal processing.
4.4. COMPLETE ENERGY DISSIPATION MODEL

State transition

Transition between different operation states is not instantaneous, as subsystems are activated and deactivated, but instead it takes a certain amount of time and an added power consumption. This section provides an explanation to the different transitions and their energy consumption.

A state transition to the sleep state is used to put the sensor into a state of hibernation after an active time period. It is used to prolong the life-time of sensor nodes when it does not need to be active. In the sleep state, a transition to the sense state is commenced when the sensor node does not need to receive or transmit any data, but needs to collect new data from its surrounding; a transition to the transmit state is needed when a sensor node needs to, for example, transmit a beacon signal; and a transition to the receive state is, for example, needed to initially find neighbours and receive data. A transition from the sense to the receive state is used by the sensor node to make certain that no other nodes are transmitting data or when it needs to act as a router; and a transition to the transmit state can be used to transmit a beacon signal or some other data. During communication, frequent transitions between transmit and receive states are executed. This happens when the sensor node needs to act as a router, establishing communication links between two or more nodes. After such a communication exchange, the sensor node either enters the sleep state or the sense state and it depends on the application. Since the frequency synthesiser is always active in both the transmit and the receive states, the transition time between these two states can be kept very short.

During a transition period, as described above, subsystems are activated or deactivated. To model this change in power consumption, the actual power consumption characteristics during a transition have been simplified with the assumption that the power consumption changes according to a linear function between the two power consumption levels at state \( i \) and \( j \) [62]. In reality, the power consumption during a transition depends on the actual components used and is different for both activation and deactivation of a subsystem. The average power consumption during a state transition can be calculated as follows

\[
P_{ij} \approx \frac{P_i + P_j}{2}
\]  

(4.17)

The transition times \( \tau_{ij} \) and \( \tau_{ji} \) usually differ, where the activation time of subsystems are usually longer than the deactivation time of the same subsystem. The total accumulated time spent in transition between operation state \( i \) and \( j \) can be calculated as follows

\[
T_{ij} = m_{ij} \tau_{ij}
\]

where \( m_{ij} \) is the number of transitions from state \( i \) to \( j \). The energy expended in the state transitions are

\[
E_{ij} = P_{ij} T_{ij} = P_{ij} m_{ij} \tau_{ij}
\]  

(4.18)

where \( P_{ij} \) and \( \tau_{ij} \) are constants for the specific sensor node.
Table 4.3. The power consumption for the four operation states.

<table>
<thead>
<tr>
<th>Operation state</th>
<th>$P_{\text{measured}}$</th>
<th>$P_{\text{datasheet}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Sleep</td>
<td>2.8 $mW$</td>
<td>1.5 $mW$</td>
</tr>
<tr>
<td>2: Sense</td>
<td>4.8 $mW$</td>
<td>4.6 $mW$</td>
</tr>
<tr>
<td>3: Transmit</td>
<td>95.3 $mW$</td>
<td>98.3 $mW$</td>
</tr>
<tr>
<td>4: Receive</td>
<td>98.2 $mW$</td>
<td>102.9 $mW$</td>
</tr>
</tbody>
</table>

Total energy consumed

The total energy consumed, $E$, in a sensor node, using this energy model, is expressed in equation (4.19) and is obtained from equations (4.12) and (4.18). The total time period can for example be five years, one month or it can be an execution cycle in an algorithm for example one round in LEACH [28]. The $E$ depends on the energy consumed in each operation state, $E_i$, and the energy expended during the state transitions, $E_{ij}$, where $i$, $j$ denotes the operation states and for this model is constrained by $1 \leq i, j \leq 4$.

$$E = \sum_{\forall i} E_i + \sum_{\forall i \neq j} E_{ij}$$

$$= \sum_{\forall i, k} P_i \tau_i(k) + \sum_{\forall i \neq j} P_{ij} m_{ij} \tau_{ij} \quad (4.19)$$

4.4.2 Measurements

To give an idea how the power consumption and the transition characteristics differ in the operation states and the state transitions. The characteristics for the Linköping sensor node [39] are presented for the operation states in Table 4.3 and for the state transitions in Table 4.4 and 4.5.

These measurements are obtained using labVIEW, a data acquisition board capable to take a sample every $5 \mu s$ and a sensor node developed by the research group in Communication Electronics at Linköping university. This sensor node contains a main board that can be connected to extension boards, which adds functionality to the main board. The major components of the main board are the ATmega 128L MCU developed by Atmel and the CC2420 radio frequency transceiver developed by Chipcon. It also contains three light emitting diodes and three crystals, which have frequencies of 32.768 KHz, 8 MHz and 16 MHz individually. The two former crystals are connected to the MCU; the first one provides a real time clock so the MCU is able to wake from power-save mode and the second one is used to generate the system clock to meet the timing requirements of the IEEE 802.15.4 MAC-layer [3][26]. The latter crystal is connected to the radio frequency transceiver so it is able to function correctly. To the main board we have connected an extension board that contains the temperature sensor DS1631 developed by Dallas semiconductor. This main and extension board provides
4.4. COMPLETE ENERGY DISSIPATION MODEL

Table 4.4. Transition times both measured and obtained from data sheets.

<table>
<thead>
<tr>
<th>From state</th>
<th>To state</th>
<th>( T_{\text{measured}} )</th>
<th>( T_{\text{datasheet}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2: Sense</td>
<td></td>
<td>( \sim 170 , \mu s )</td>
<td>–</td>
</tr>
<tr>
<td>1: Sleep</td>
<td>3: Transmit</td>
<td>( \sim 2.68 , ms )</td>
<td>1.892 , ms</td>
</tr>
<tr>
<td>4: Receive</td>
<td></td>
<td>( \sim 2.65 , ms )</td>
<td>1.892 , ms</td>
</tr>
<tr>
<td>1: Sleep</td>
<td></td>
<td>( \sim 100 , \mu s )</td>
<td>–</td>
</tr>
<tr>
<td>2: Sense</td>
<td>3: Transmit</td>
<td>( \sim 2.68 , ms )</td>
<td>1.892 , ms</td>
</tr>
<tr>
<td>4: Receive</td>
<td></td>
<td>( \sim 2.58 , ms )</td>
<td>1.892 , ms</td>
</tr>
<tr>
<td>1: Sleep</td>
<td></td>
<td>( \sim 120 , \mu s )</td>
<td>–</td>
</tr>
<tr>
<td>3: Transmit</td>
<td>2: Sense</td>
<td>( \sim 80 , \mu s )</td>
<td>–</td>
</tr>
<tr>
<td>4: Receive</td>
<td></td>
<td>–</td>
<td>192 , \mu s</td>
</tr>
<tr>
<td>1: Sleep</td>
<td></td>
<td>( \sim 110 , \mu s )</td>
<td>–</td>
</tr>
<tr>
<td>4: Receive</td>
<td>2: Sense</td>
<td>( \sim 80 , \mu s )</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>3: Transmit</td>
<td>–</td>
<td>192 , \mu s</td>
</tr>
</tbody>
</table>

...the necessary hardware to make power and time measurements. The data sheet for the components provides the reference data for these measurements when available.

The power consumption measurement for each operation state was obtained by measuring the current drawn from the battery. The current in the sensor node was obtained by taking the average current drawn over a one minute period. This provides us with a rather good approximation for the current drawn from the battery. Since the supply voltage provided by the voltage regulator in the sensor node has an output voltage of 3.3V. The power dissipation is easily calculated from \( P = UI \) where \( U \) is the output voltage of the voltage regulator and \( I \) is the current drawn from the battery. The power consumption measured and achieved from the different data sheets can be viewed in Table 4.3. The power consumption values obtained from the data sheets only takes the MCU, the radio frequency transceiver, voltage regulator and the sensor into account.

The time duration to complete a state transition occurs when the current drawn in the first operation state changes to the second one. To be able to measure these transition times, we need to define a start and a stop condition for a transition. We first define an interval, \( I_i = [A_i, B_i] \), for operation state \( i \) where \( A_i \) is the minimum current for operation state \( i \) and \( B_i \) is the maximum current drawn for operation state \( i \). During a transition from operation state \( i \) to operation state \( j \) the start of the transition is defined as the time instant when the current deviates from the interval, \( I_i \), and the stop condition is similarly defined as the time instant when the current is in the interval, \( I_j \). The measured transition times and those obtained from the data sheets are show in Table 4.4.

It is important to remember that there are measurement errors related to both the measured power and time. These errors are mainly related to accuracy and should be considered when using the presented measurements and evaluating the results pre-
CHAPTER 4. ENERGY DISSIPATION MODELS

Table 4.5. Energy consumed in a transition for calculated and measured values.

<table>
<thead>
<tr>
<th>From state</th>
<th>To state</th>
<th>$E_c$</th>
<th>$E_m$</th>
<th>$E_c/E_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2: Sense</td>
<td>1: Sleep</td>
<td>0.64 $\mu$J</td>
<td>0.67 $\mu$J</td>
<td>0.96</td>
</tr>
<tr>
<td>3: Transmit</td>
<td>4: Receive</td>
<td>131.5 $\mu$J</td>
<td>122.0 $\mu$J</td>
<td>1.08</td>
</tr>
<tr>
<td>4: Receive</td>
<td>1: Sleep</td>
<td>133.8 $\mu$J</td>
<td>124.7 $\mu$J</td>
<td>1.07</td>
</tr>
<tr>
<td>3: Transmit</td>
<td>2: Sense</td>
<td>5.9 $\mu$J</td>
<td>5.1 $\mu$J</td>
<td>1.16</td>
</tr>
<tr>
<td>4: Receive</td>
<td>1: Sleep</td>
<td>134.1 $\mu$J</td>
<td>123.1 $\mu$J</td>
<td>1.09</td>
</tr>
<tr>
<td>2: Sense</td>
<td>3: Transmit</td>
<td>18.6 $\mu$J</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>4: Receive</td>
<td>1: Sleep</td>
<td>4.0 $\mu$J</td>
<td>3.7 $\mu$J</td>
<td>1.08</td>
</tr>
<tr>
<td>3: Transmit</td>
<td>2: Sense</td>
<td>5.6 $\mu$J</td>
<td>4.9 $\mu$J</td>
<td>1.14</td>
</tr>
<tr>
<td>4: Receive</td>
<td>3: Transmit</td>
<td>4.1 $\mu$J</td>
<td>3.9 $\mu$J</td>
<td>1.05</td>
</tr>
<tr>
<td>2: Sense</td>
<td>4: Receive</td>
<td>18.6 $\mu$J</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

In this section the complete energy dissipation model is evaluated. The evaluation of the transition states is important. This is due to the simplifications done in the calculations, which might lead to errors in the estimated energy consumption. The measured energy consumption is thus compared to that of the simplified model. After this evaluation a small scenario is evaluated in which all operation states and the corresponding state transitions are used. This evaluation shows how the proposed energy model reflects reality.

**Transitions**

In each transition in the proposed energy model there is a simplification done, which uses the expected power consumption over the entire transition period to obtain the energy consumed during a transition, see Equation (4.17). This estimated value is here compared to the actual energy consumed. The measured power is taken every $5 \mu$s and as such the area for the measured energy is obtained by adding the consecutive rectangular regions for a state transition. The start and stop condition for the state transitions are obtained in the same way as in subsection 4.4.2.

As can be seen from Table 4.5 the difference is at most 16% during a transition between the energy calculated $E_c$ and the energy measured $E_m$. As mentioned before, this is a simplistic model of the state transitions and to obtain calculated values that have an average that are less than 17% of the measured values are good. This is also better than we expected from the transition model. This table also shows that the en-
4.4. COMPLETE ENERGY DISSIPATION MODEL

Figure 4.6. Measured values for the scenario.

Energy consumed during a transition from the sleep to the sense state are underestimated, where as the energy for the other state transitions are overestimated. In this case an overestimation of the energy consumption is preferable to an underestimation, as it provides a more pessimistic view of the energy consumption characteristics.

Scenario

The scenario used to evaluate the proposed energy model has a duration of 20830 ms and it is divided into six time periods, see Figure 4.6. The first time period is the sleep period with a duration of 8 s and this period is followed by a 1.6 s long sense period. Following the sense period is a 8 s long sleep period which is in turn changed to a receive period with a duration of 1.6 s. The fifth part is a 30 ms long transmit period which is then changed to the final sleep period with a duration of 1.6 s. In addition to evaluating the complete scenario we have also chosen to evaluate three subintervals. These subintervals are defined as follows: The first subinterval is from 7 to 10 s, the second is between 12 and 14 s and the last one is from 17 to 20 s.

From Table 4.6 it can be seen that for the complete scenario the calculated energy and the measured energy is roughly the same. The calculated values are, however, a slight underestimation of the measured once. It can also be seen that subinterval 1 provides a good estimate of the measured values for the combination of sleep and sense states as well as the state transitions. Even though there are two peaks in the sense state. It can also be seen that subinterval 2 overestimates the energy consumed in the sleep state and that subinterval 3 underestimates the energy consumed for the combination of sleep, receive and transmit states as well as the state transitions. This
**Table 4.6.** The calculated and measured energy consumption for the scenarios.

<table>
<thead>
<tr>
<th>Subinterval</th>
<th>$E_c$</th>
<th>$E_m$</th>
<th>$E_c/E_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[7, 10]$</td>
<td>11.6 mJ</td>
<td>11.6 mJ</td>
<td>1.00</td>
</tr>
<tr>
<td>$[12, 14]$</td>
<td>5.6 mJ</td>
<td>5.4 mJ</td>
<td>1.04</td>
</tr>
<tr>
<td>$[17, 20]$</td>
<td>164.0 mJ</td>
<td>165.4 mJ</td>
<td>0.99</td>
</tr>
<tr>
<td>Complete scenario</td>
<td>217.3 mJ</td>
<td>218.0 mJ</td>
<td>1.00</td>
</tr>
</tbody>
</table>

**Figure 4.7.** The sensor node topology for the environmental control scenario.

can be due to the peaks obtained during transitions between these states, as seen in Figure 4.6. For these measured values this energy dissipation model provides good estimates that can be used to evaluate the energy dissipation in wireless sensor nodes or networks.

### 4.5 Case study

In this section a case study is presented for environmental control in an office building, where sensor nodes are used to gather both event-based and periodic data. In this study we use the complete energy dissipation model to calculate the average energy consumed during one beacon interval for a cluster of sensor nodes.

Vision a building, where the temperature, humidity and other parameters are gathered by small sensor nodes. These nodes are positioned both inside and outside the building. All the measurements are then transmitted wirelessly to the environmental control centre using a backbone of sensor nodes.

The wireless sensor network is organised into clusters where each cluster head
is a part of the backbone in the network. In this structure each cluster forms a star topology with the cluster head as the centre and all of its associated sensor nodes as endpoints, see Figure 4.7. The Medium Access Control (MAC)-layer used for all these sensor nodes is the IEEE 802.15.4, which is designed to be used in energy constrained wireless sensor networks. It provides a bit-rate of 250 kbps. In each cluster the cluster head’s main function is to gather the collected data from sensor nodes and to relay data through the backbone to the environment control centre. All the cluster heads are power-rich so no communication problems related to the energy level can occur.

The beacon interval, which is the time duration between two consecutive synchronisation beacons, has a duration of 31457 ms and is divided into an active and an inactive portion, see Figure 4.8. The active portion of the beacon interval, see Figure 4.9, continues for the rest of the superframe which has a duration of 15.36 ms. A superframe consists of sixteen time slots divided into a contention access period (CAP) and a contention free period (CFP). In the CAP both control information and gathered data may be transmitted to the nodes and the cluster head. In the CFP information gathered by the sensor nodes (temperature, humidity, emitted light, movement, etc.) are transmitted to the cluster head. The time slots have a duration of 0.96 ms each and the beacon is transmitted in the first time slot in the superframe. The inactive portion, see Figure 4.10, is the remainder of the beacon interval duration and starts with a transition period of length 0.115 ms. That is a state transition from either the transmit or the receive state to the sleep state. The state transition time is obtained as the mean of a state transition from the transmit or the receive state to the sleep state for simplicity. After this transition there are ten sub intervals of length 3143.17 ms that
are divided into a sleep, a transition and a sense part. The sleep part has a duration of 3000 ms, the transition parts duration is 0.17 ms and the sense part is the remaining 143 ms. Between each of these sub-intervals a transition occurs from the sense to the sleep state with a duration of 0.09 ms. When the last sub interval has completed a transition from the sense to the receive state takes place with a duration of 2.6 ms. In the remaining 6.4 ms the sensor node is in the receive state and continues to gather data about its surrounding while it waits for the next synchronisation beacon from the cluster head.

For the presented scenario we are going to evaluate the power consumption for the sensor nodes in one cluster. In this cluster there are four battery powered sensor nodes statically placed at different location to gather information for the control centre about deviating data and a cluster head which function is to collect data from the sensor nodes and relay it to the control centre. The superframe, as mentioned above, contains sixteen time slots where four are assigned to the collision free period. For simplicity, the data received are not acknowledged by an acknowledgement frame by either the sensor nodes or the cluster head. In the collision access period the cluster head schedules the transmission order for the sensor nodes by broadcasting one packet containing all information about the transmission time schedule in the collision free period. The cluster head transmits the broadcast packet in one time slot that is uniformly chosen from the eleven time slots available in the collision access period. Since the first timeslot is used for the beacon signal. Here we assume that if no collisions occur the broadcast packet is successfully received by all sensor nodes; otherwise a collision occurs and no node will receive the new transmission schedule for the collision free period of the superframe. In that case the sensor nodes use the previous transmission schedule in the collision free period again. If the data that a sensor node transmits contains more bits than that of one packet, the data is spread out into a maximum of two packets. One of these packets contains the more important data and is transmitted in the collision free period and the other packet is transmitted in the collision access period.
collision the transmitted packets are not retransmitted for simplicity.

The probability that a sensor node needs to transmit one packet in the collision access period during a beacon interval is $p$ and the time slot in which the packet is transmitted is uniformly distributed among the twelve time slots available. The probability for $n$ sensor nodes in one cluster to transmit $l$ packets together in the collision access period is drawn according to the binomial distribution as follows

$$ p_l = Pr\{X_n = l\} = \binom{n}{l} p^l (1 - p)^{n-l}, \quad (4.20) $$

where $X_n = X_1 + X_2 + \cdots + X_n$ and $X_i, 1 \leq i \leq n$, is a Bernoulli random variable. The probability that the same node transmits in the last time slot of the collision access period, event $A$, and the first time slot of the collision free period, event $B$, is $p_{ab} = Pr\{A \cap B\} = Pr\{A\} Pr\{B\}$ as these events are independent of each other. However, for simplicity we neglect $p_{ab}$, since its impact on the result is small. From this, we are able to calculate the average time duration in the operation states and the state transitions for one cluster. The average time duration in the transmit state for the entire cluster is obtained as

$$ T_3 = \sum_{l=0}^{n} l p_l \tau_{slot} + n \tau_{slot}, \quad (4.21) $$

where $\tau_{slot}$ is the duration of one timeslot. The average time duration in the receive state during the active portion of the beacon interval is as follows

$$ T_{4,ac} = 16n \tau_{slot} - T_3 - T_{trans,ac} $$

$$ = \left(15n - \sum_{l=0}^{n} l p_l\right) \tau_{slot} - T_{trans,ac}, \quad (4.22) $$

where $T_{trans,ac}$ is the time duration of all state transitions between the transmit and the receive state in the active portion of the beacon interval, see Equation (4.23). The time duration in the inactive portion of the beacon interval is $T_{4,iac} = n \tau_{4,iac}$, where $\tau_{4,iac}$ is the time that one sensor node is in the receive state during the inactive portion of the beacon interval. Thus, the total time in the receive state is easily obtained as $T_4 = T_{4,ac} + T_{4,iac}$. In both the sense and the sleep state, the time for one visit is equal to the time for all other visits to the same state, that is $\tau_2 = \tau_2(k) \forall k$ and $\tau_1 = \tau_1(k) \forall k$. Since there are ten sub intervals, $r$, in the inactive portion of the beacon interval the total time for this cluster in the sense state is $T_2 = nr \tau_2$ and in the sleep state is $T_1 = nr \tau_1$.

Each state transition between operation states $i$ and $j$ has a constant duration and thus it suffices to calculate the number of transitions between each operation state. The number of transitions in the inactive part of the beacon interval can easily be obtained by counting: from the sleep to the sense state, $m_{12} = 10n$; from the sense to the sleep state, $m_{21} = 9n$; from the transmit to the sleep state, $m_{31} = 1$; form the receive to the
CHAPTER 4. ENERGY DISSIPATION MODELS

sleep state, \( m_{41} = n - 1 \); from the sense to the receive state, \( m_{24} = n \). The number of state transitions in the active part of the beacon interval depends on the number of transmissions and as the transition duration and the power consumed between the transmit and the receive state is the same i.e. \( \tau_{34} = \tau_{43} \) and \( P_{34} = P_{43} \). The duration of these transitions can be calculated at the same time and is obtained from

\[
T_{\text{trans,ac}} = \left( \sum_{l=0}^{n} 2lp_l + (2n - 1) \right) \tau_{\text{trans}}
\]

(4.23)

where \( \tau_{\text{trans}} = \tau_{34} = \tau_{43} \).

By calculating the time duration in the operations and the state transitions for all sensor nodes in a cluster over one beacon interval, the energy consumed follows easily from Equation (4.19) and \( E \approx 370.1 \text{ mJ} \) is the total energy consumed in a cluster. This value of \( E \) was obtained by assuming that \( p = 0.1 \) in Equation (4.20).

4.6 Conclusion

In this chapter three energy dissipation models have been described, namely the radio energy dissipation model, the state based energy dissipation model and the complete energy dissipation model. These models have been presented in order of complexity, in terms of which subsystems that are taken into account during modelling of a sensor nodes energy consumption.

The first model only takes into account the communication subsystem, which it models rather accurately. However, other parts of the sensor node are neglected. As such, this energy models can only be used to compare the energy consumption of infrastructure and application communication between sensor nodes. Unfortunately, this limits its usefulness for evaluation of the energy efficiency in other types of protocols and algorithms.

The second model uses in addition to the communication subsystem also the sensor subsystems to model sensor nodes energy dissipation. However, the processing subsystem is neglected, which consumes much energy. The activity of the processing subsystem is closely related to the activity in both the sensing and the communication subsystems. Since, parts of the communication stack are executed by the processor and the sensed information needs to be processed before it is forwarded. Thus, by neglecting the processing subsystem a large underestimation of the energy consumed in a sensor node will occur.

To alleviate the problems of the previously presented energy models, the complete energy dissipation model is based on the basic sensor node architecture and as such it includes all fundamental subsystems i.e. the communication, the processing and the sensor subsystem. This provides a good accuracy in term of the energy dissipation in a sensor node. To further improve its usefulness the state transitions are also included such that the scheduling between different operations states can be decided based on its energy overhead and the expected time to the next event.
4.6. CONCLUSION

This chapter also includes measurements on an actual sensor node for the complete energy dissipation model. These measurements include the consumed power in the operation states and the state transitions. It also includes the time it takes to complete a state transition between two operation states. These values are used in the evaluation of the complete energy dissipation model.

In addition to alleviating the problems of the energy models presented in sections 4.2 and 4.3. The complete energy dissipation model is compared to a typical sensor node. The results are promising in which the calculated energy consumption for the scenario used is less than 1% of the measured value. However, the energy model provides a small underestimation of the energy consumed. It also shows that the energy consumed during a state transition is less than 17% of the measured energy consumption and that the energy is usually a little overestimated.
We start this chapter with a short description of the definition of a dominating set, connected dominating set and how broadcast is conducted at the MAC-layer in a network. Then we describe quite generally the idea behind flooding and present classic and controlled flooding in Section 5.2. The former lets all nodes retransmit a broadcast message and the latter tries to reduce the number of rebroadcast nodes. Following this are two very general classifications of controlled flooding protocols, namely the neighbour knowledge class in Section 5.3 and the non-neighbour knowledge class in Section 5.4. In each of these two categorise, two protocols are more thoroughly reviewed. Then two controlled flooding protocols are proposed, which make their rebroadcast decisions based on the received signal strength at the receiving node. Following is a comparison between four of the presented controlled flooding protocols, as reference classic flooding is used. Finally, some concluding remarks are presented.

5.1 Preliminaries

This section presents dominating sets and how broadcast at the MAC-layer is commenced. The former is used to determine the minimum number of transmitting nodes needed to cover all nodes in a network. Notice that this does not necessarily form a rooted-tree. However, to form a rooted three a connected dominating set needs to be used in the network. This determines the number of nodes required to broadcast a message, which will be received by all nodes. This is under the assumption that the transmission distance does not vary and that no errors can occur. The latter describes how IEEE 802.11 and 802.15.4 broadcast is conducted and why it cannot use the same approach as that of unicast.
CHAPTER 5. EFFICIENT FLOODING PROTOCOLS

5.1.1 Dominating set

To be able to define a dominating set in a network, we first need to present the network model and some assumptions related to that model. Assume that there are $N$ static nodes using omnidirectional antennas in a geographic area and these nodes are connected to each other using wireless transmission links. A link exists between two nodes if these nodes can receive each other’s transmissions, i.e. a bidirectional link. These nodes are also neighbours of each other. This type of network can be modeled as a graph $G(V, E)$, where $V$ represent the vertex set (i.e. the nodes) and $E$ is the edge set (i.e. the transmission links). Furthermore, assume that the transmissions over the wireless channel are lossless and that $G(V, E)$ is connected. To reach all nodes in a network only a subset of the nodes in $V$ needs to transmit the message. This subset is referred to as a dominating set of the network. A dominating set in a graph is formally defined below.

**Definition 5.1** In a graph $G(V, E)$, a set $S \subseteq V$ is a **dominating set** if every vertex not in $S$ have a neighbour in $S$. The minimum size of a dominating set in $G$ is referred to as the **dominating number** $\gamma(G)$.

To constrain the dominating set even more the definition for a connected dominating set follows.

**Definition 5.2** A dominating set $S$ in a graph $G(V, E)$ is a **connected dominating set** if every vertex in $S$ have at least one neighbour also in $S$.

To get a better understanding of a dominating set and a connected dominating set, an illustration is shown in Figure 5.1. Graph $G$ in this figure shows two dominating sets. The one with black vertices is a minimum dominating set of 3 vertices (i.e. $\gamma(G) = 3$) and the one with squares around their vertices is a connected dominating set of size 6.
5.2 Flooding

Flooding protocols are used as basic building block for more advance network-layer protocols, like routing protocols and cluster construction and management schemes. Brad Williams et. al [70] have categorised the different flooding protocols into four categories: Simple flooding, Probability-based methods, Area-based methods and Neighbour knowledge methods. The first category of flooding protocols include only the classic flooding protocol and it is explained in subsection 5.2.1. The other three categories are referred to as controlled flooding protocols and the fundamentals are briefly described in subsection 5.2.2. The categories used in this thesis are similar to the one presented above and this categorisation is illustrated in Figure 5.2.

5.2.1 Classic flooding

Classic flooding is one of the simplest ways to relay a message from a source node to all other nodes in a network. This algorithm works as follows. The source node broadcasts a message to all of its neighbours; each of these neighbours processes the message and rebroadcasts the message exactly once; this continues until the broadcast message has been received by all nodes in the network. The rebroadcast nodes try to process and relay the message as fast as they can i.e. without any delay. However, the time it takes to accomplish this depends on a node’s processing speed of the message, access delay to the channel and wait time for other messages in the queues.
The main problems when using this algorithm include unproductive and often harmful bandwidth congestion and inefficient use of nodes resources. The former problem relates to the rebroadcast of messages from nodes in close proximity (i.e. within transmission range) to each other at approximately the same time. This results in collisions and contention of the wireless medium. The latter problem relates to the transmission redundancy of a message, in which most nodes already have received the same broadcast message previously. This is due to the broadcast nature of the wireless channel. These two problems are usually referred to as the broadcast storm problem.

5.2.2 Controlled flooding

To address the broadcast storm problem in wireless networks, researchers have proposed different techniques to limit the number of rebroadcasts of a message in the network. These techniques are commonly referred to as controlled or efficient flooding schemes.

It is known that the problem to reliably broadcast a message with minimum energy consumption is equivalent to finding a minimum connected dominating set (MCDS) in a lossless wireless network. Thus, a source node that broadcasts a message only need to transmit it to all nodes in the MCDS and of these nodes forwards the received message. From the connectivity of the MCDS it is obvious that all nodes in the MCDS receive the message. Since all nodes in the MCDS are dominating the other nodes, the dominated nodes also receive that message. Due to that the number of nodes in a connected dominating set are minimal. The number of transmitted messages in the network is also minimal. This obviously results in minimal energy consumption when only considering the number of transmissions. However, due to the complexity in determining a MCDS, which is a NP-hard problem when the entire topology is known. Therefore, reliable and energy-efficient broadcasting is a challenging problem in wireless sensor networks.

Due to the computational complexity of a MCDS, heuristic approaches that ap-
5.2. FLOODING

Algorithm 1: Skeleton algorithm for controlled flooding protocols.

<table>
<thead>
<tr>
<th>Data: message $m$ is received at node $A$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 if message $m$ has been received before then</td>
</tr>
<tr>
<td>2 message $m$ is dropped;</td>
</tr>
<tr>
<td>3 else</td>
</tr>
<tr>
<td>4 node $A$ initiates a timer;</td>
</tr>
<tr>
<td>5 wait until timer expires;</td>
</tr>
<tr>
<td>6 if retransmit message $m$ then</td>
</tr>
<tr>
<td>7 node $A$ broadcasts message $m$;</td>
</tr>
<tr>
<td>8 else</td>
</tr>
<tr>
<td>9 node $A$ drops message $m$;</td>
</tr>
<tr>
<td>10 end</td>
</tr>
<tr>
<td>11 end</td>
</tr>
</tbody>
</table>

proximate the MCDS have been proposed. The heuristic approaches available can be divided into either centralised or distributed schemes. The former schemes are not a viable alternative in wireless sensor networks. This is due to the large control overhead that is introduced in the network and the large memory requirement needed for the sensor nodes. The latter schemes provide a more appropriate solution for wireless sensor networks, since they have lower control overhead and memory requirement compared to the centralised schemes. The remaining parts of this chapter only take into account the distributed heuristic approaches.

General approach

The distributed schemes usually employ a heuristic approach that is similar to the skeleton in Algorithm 1 [63]. With this general approach, protocols that utilise this simple idea differ in at least two design aspects. These aspects are

1. How should the delay be assigned to a message $m$ at node $A$?

2. How does node $A$ determine if message $m$ is going to be rebroadcast or dropped?

The first aspect of the two has a simple solution, in which the rebroadcast node assigns a random delay in the interval $[0, \text{max\_wait\_time}]$ before retransmitting the message. However, this strategy would be good if the retransmission order is of no importance. The importance of the retransmission order depends on how a node decides if it should retransmit the message or not (the second aspect). If a retransmitting node drops a packet with probability $p$ then the retransmission order is of no importance. In this case a random delay can be used to differentiate the retransmission time of the nodes to lower the contention and the number of collisions in the channel. On the other hand, if a distance based discarding policy is used. A node that is positioned close to the maximum transmission range would preferably retransmit the message before a
node that is positioned closer to the transmitting node. In this case the retransmission order is of importance and a dynamic delay function based on the transmission distance is preferable to a random delay for the retransmitting nodes.

The second aspect relates to how a node estimates a redundant broadcast message and how it accumulates the knowledge to assist in that decision. This results in two very different classes of protocols, namely neighbour knowledge and non-neighbour knowledge. To make a rebroadcast decision the former relies on information about the topology in vicinity to the deciding node. The neighbour knowledge is usually in a 1-hop or 2-hop radius from the deciding node, see Figure 5.3. The latter category usually bases its decision on either a probabilistic or an area-based measure. The probabilistic measure can for example be as simple as setting a rebroadcast probability $p$ for each node. This means that on the average $Np$ nodes rebroadcast the message, where $N$ is the number of nodes in the network. If $p = 1$, then it is equal to classic flooding. The area-based measure usually uses the additional coverage area to assist in its rebroadcast decisions. Both classes of methods are further discussed in section 5.3 and 5.4.

5.3 Neighbour knowledge methods

This class of methods [45][53][69][70][38][31][51] rely on neighbour knowledge to assist in the decision to determine if a message is redundant. The neighbour information at a node has a scope of $n$-hops. This means that a node collects topology information within a radius of $n$-hops, see Figure 5.3 that shows all nodes within a radius of 2-hops. To collect topology information in a larger radius than 2-hops leads
to a very large increase in the overhead and the time to complete the topology update procedure increases. As such, most neighbour knowledge protocols use either 1-hop or 2-hop neighbour knowledge. The simplest of these protocols uses 1-hop neighbour knowledge, which is acquired through the use of periodic beacon messages. These beacon messages are usually referred to as hello messages and they allow all nodes to be aware of their neighbours. The more advance protocols rely on 2-hop neighbour knowledge, in which each node add their neighbours to the hello message that they transmit. By doing so, each node in the network will know who are its closest neighbours and further who are their neighbours. From this information the nodes make a decision to rebroadcast the message or not. This will form a connected dominating set (CDS) that covers all nodes connected to the source node.

The neighbour knowledge class of algorithms can further be divided into a neighbour-designating and a self-pruning category. In the former category the transmitting node decide the forwarding status of the receiving nodes, i.e. to rebroadcast or to drop the message. In the latter category of methods it is the receiving node itself that decides its forwarding status, either to rebroadcast the message or not.

One advantage with these protocols is generally better performance. This is due to the neighbour knowledge that allows nodes to make better rebroadcast decisions. However, in wireless sensor networks this category of protocols has a number of disadvantages. This includes memory requirements (e.g. storage of neighbour tables) and communication overhead (e.g. beacon messages, neighbour information added to the broadcast message, etc.). From this point of view a more resource efficient scheme is preferable. However, it is important to notice that there is a trade-off between performance and resource efficiency.

Below follows a description of two protocols in the self-pruning category, namely flooding with self-pruning and scalable broadcast algorithm. Flooding with self-pruning rely on 1-hop neighbour knowledge to make its rebroadcast decisions. On the other hand the scalable broadcast algorithm makes its rebroadcast decision based on the use of 2-hop neighbour knowledge instead of 1-hop neighbour knowledge.

5.3.1 Flooding with self-pruning

The flooding with self-pruning protocol [38] uses 1-hop neighbour knowledge information to reduce the number of transmissions required to do a network wide broadcast i.e. flooding. The neighbours $N(v_i)$ of a node $v_i \in V$, where $V$ is the nodes in the network, is obtained by exchanging periodic beacon messages. When a node $A$ initiates a broadcast it adds its neighbours $N(A)$ to the broadcast message. Another node $B$ that receives the broadcast message compares its neighbours $N(B)$ to that in the received broadcast message and the transmitting node, in this case $N(A)$ and $\{A\}$. If no additional nodes can be covered, node $B$ refrains from doing a rebroadcast; otherwise a rebroadcast occurs.

The rebroadcast procedure of this protocol is derived more formally below.
1. Source node \( S \) adds its neighbour list \( N(S) \) to message \( m \). Node \( S \) then broadcasts message \( m \) to all of its neighbours. If node \( S \) receives message \( m \) later on it is dropped.

2. For a node \( v_i \) when it receives a broadcast message \( m \) from a node \( v_j \), it performs the following operations:

   If message \( m \) has been transmitted before or \( N(v_i) \subseteq N(v_j) \cup \{v_j\} \), then message \( m \) is dropped.

   Else add \( N(v_i) \) to message \( m \) and rebroadcast it.

This protocol checks if the neighbour set in the first received message covers its own neighbour set. If additional nodes can be reached it directly rebroadcasts the message. It would be better to wait a random time before a rebroadcast of the message occurs to collect neighbour knowledge information from redundant messages. This can aid in the rebroadcast decision and at the same time lower the number of collision, the number of redundant messages and the contention of the wireless channel. This is the idea for the improved flooding with self-pruning (iFSP) protocol.

The rebroadcast procedure for this protocol is formally derived below

1. Source node \( S \) adds its neighbour list \( N(S) \) to message \( m \). Node \( S \) then broadcasts message \( m \) to all of its neighbours. If node \( S \) hears message \( m \) again it is discarded.

2. For a node \( v_i \) when it receives a broadcast message \( m \) from a node \( v_j \), it performs the following operations:

   If message \( m \) has been transmitted before or \( N(v_i) \subseteq N(v_j) \cup \{v_j\} \), then message \( m \) is discarded, the timer is cancelled and duplicate messages received later on will be dropped.

   Else if message \( m \) is a new broadcast message, then let \( K(v_i, m) = N(v_j) \cup \{v_j\} \) and initiate a countdown timer. Set it to a random value in the interval \([0, max\_wait\_time]\). This is done to delay the rebroadcast operation for a random period of time.

   Else update \( K(v_i, m) = K(v_i, m) \cup N(v_j) \cup \{v_j\} \) and discard message \( m \).

   When the countdown timer expires, if \( N(v_i) \subseteq K(v_i, m) \), then the rebroadcast is prohibited; otherwise, \( N(v_i) \) is added to message \( m \) and a rebroadcast message \( m \) occurs.

To get a better idea how the latter protocol operates, an example follows. The network topology for this example is illustrated in Figure 5.4. A node \( S \) initiates the iFSP protocol by a broadcast of message \( m \) and adds \( N(S) = \{A, B\} \), i.e. the neighbour list of node \( S \) to message \( m \). Both nodes \( A \) and \( B \) receives the broadcast and calculates \( K(A, m) = K(B, m) = \{A, B, S\} \). Then they calculate the delay and
assigns it to the countdown timer. Lets assume that the timer at node $A$ expires first. Since $N(A) \not\subseteq K(A, m) = \{S, B, C\} \not\subseteq \{A, B, S\}$, it adds $N(A) = \{S, B, C\}$ to message $m$ and rebroadcasts it. Nodes $S$, $B$ and $C$ receives the broadcast. Node $S$ drops message $m$, since a broadcast of message $m$ has already been done and node $B$ also drops message $m$, since $N(B) \subseteq N(A) \cup \{A\} = \{S, A, C\} \subseteq \{S, A, B, C\}$. Node $C$ determines that $N(C) \subseteq N(A) \cup \{A\} = \{A, B\} \subseteq \{S, A, B, C\}$ and drops the message.

Another scheme is the prioritised flooding with self-pruning protocol [31] that have been proposed in the literature. Instead of a random delay it uses a dynamic delay function that bases the assigned delay on the number of additional nodes covered. This protocol can be seen as a combination of the counter-based scheme [45] and that of the scalable broadcast algorithm [51]. The latter is presented below.

### 5.3.2 Scalable broadcast algorithm

The basic idea of this algorithm is the same as that of flooding with self-pruning. However, the scalable broadcast algorithm (SBA) [51] uses 2-hop neighbour knowledge instead of 1-hop neighbour knowledge. Based on this knowledge a message is dropped if all its neighbours have been covered by previous broadcasts; otherwise it rebroadcasts the message. To do this, the node needs to determine the transmitter of the received message, which it assumes can be accomplished. From this information it uses its 2-hop neighbour knowledge to determine which nodes have been covered by the received broadcast message. These nodes are added to the cover set of this particular message. Simply, a node rebroadcasts the message if the cover set does not cover its neighbouring nodes entirely; otherwise the message is dropped.

The delay function used in this algorithm lets nodes with more neighbours rebroadcast the message with a shorter delay than a node with fewer number of neighbours. The reason for this is that more nodes can be covered in a shorter time and using fewer
number of transmissions. The functions to calculate the delay $D$ is defined as follows

$$T_0 = \frac{1 + d_m(v_i)}{1 + d(v_i)}$$

$$D = U(\Delta T_0),$$

(5.1)

where $d(v_i) = |N(v_i)|$ (i.e. the degree of node $v_i$), $d_m(v_i)$ is the maximum degree of $N(v_i)$, $\Delta$ is a small constant delay and $U(x)$ is a function that returns a random number uniformly distributed in the interval $[0, x]$.

The rebroadcast procedure of this protocol is derived below.

1. Source node $S$ broadcasts message $m$ to all of its neighbours. If node $S$ receives message $m$ at a later time it is dropped.

2. For a node $v_i$ when it receives a broadcast message $m$ from a node $v_j$, it performs the following operations:

   If message $m$ has been transmitted before or $N(v_i) \subseteq N(v_j) \cup \{v_j\}$, then message $m$ is dropped, the timer is cancelled and duplicate messages received later on will be dropped.

   Else if message $m$ is a new broadcast message, then let $K(v_i, m) = N(v_j) \cup \{v_j\}$ and initiate a countdown timer. Set it to the value generated by equation (5.1). This is done to delay the rebroadcast operation depending on the node degree.

   Else update $K(v_i, m) = K(v_i, m) \cup N(v_j) \cup \{v_j\}$ and discard message $m$.

   When the countdown timer expires, if $N(v_i) \subseteq K(v_i, m)$, then the rebroadcast is prohibited; otherwise, a rebroadcast of message $m$ occurs.

To get a better understanding of how this protocol operates, a short example follows. Source node $S$ needs to distribute some information to another node in the network, see Figure 5.5. However, it does not know where that node is located. To locate that node, node $S$ initiates SBA by a broadcast of message $m$. The nodes $A, B$
and $E$ receives broadcast message $m$. From message $m$ the receiving nodes know that it originates from node $S$. Node $E$ discards message $m$, since $N(E) \subseteq N(S) \cup \{S\}$ (i.e. $\{S\} \subseteq \{A, B, E, S\}$). Both nodes $A$ and $B$ calculate its cover set, which for the two nodes are $K(A, m) = K(B, m) = N(S) \cup \{S\} = \{A, B, E, S\}$. Since these two nodes have the same degree and the highest degree of neighbours, delay function (5.1) randomly generates a number in the same range. Lets assume that the countdown timer expires first at node $A$. Node $A$ then rebroadcasts message $m$, which is received by nodes $S$, $B$ and $C$. Node $S$ drop message $m$, since it has already transmitted it and node $B$ also drops the message, since $N(B) \subseteq N(A) \cup \{A\}$ (i.e. $\{S, A, C\} \subseteq \{S, C, B, A\}$). Node $C$ drops message $m$, since $N(C) \subseteq N(A) \cup \{A\}$ (i.e. $\{A, B\} \subseteq \{B, C, S, A\}$).

5.4 Non-neighbour knowledge methods

This class of protocols [64][45][21][70][59][47][50] can be further divided into probability-based and area-based methods. The former category relies on a probability measure to determine its rebroadcast nodes. A very simple approach is for a node to rebroadcast a message with a fixed probability $p$, which is done after a small random delay. This results in $pN$ rebroadcast nodes, where $N$ is the number of nodes in the network. The goal of the latter category is to cover the entire geographic area with as few broadcasts as possible. As such, each broadcast tries to cover as much new geographic area as possible. Thus, the main obstacles for this category of methods are to determine the additional coverage area of a node and to maximise the additional coverage area without the use of neighbour knowledge.

The advantages to use this class of flooding protocols in a wireless sensor network are that they are relatively simple, have a low communication overhead (e.g. no periodic beacon signal) and a low memory requirement (e.g. no neighbour knowledge). However, this also results in a performance loss due to that no neighbour knowledge is used. An additional disadvantage for the area-based methods are that the location information needs to be obtained in some manner.

Below follows a description of two area-based methods, namely the location-based scheme and the optimised broadcast protocol. The location-based scheme determines if a node should retransmit a broadcast message based on the additional coverage. The optimised broadcast protocol, on the other hand, determines optimal rebroadcast locations, based on the coverage of hexagons.

5.4.1 Location-based scheme

The location-based scheme [45] requires that each node knows its own location and each broadcast message includes the location of the transmitting node. When a node receives a broadcast message it calculates the additional area that can be covered if this node rebroadcasts the message. Let $AC((x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k))$ denote
the additional coverage divided by $\pi d^2$, where $k$ is the number of received broadcasts from the nodes located at the following positions $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$. The additional coverage $AC$ is then compared to a predefined coverage threshold $T_a$, which is assigned a value in the interval $[0, 0.61]$. If $AC \geq T_a$ the receiving node rebroadcasts the message; else it is discarded.

The broadcast procedure of this scheme is formally derived below.

1. Source node $S$ adds its own location to message $m$ and then it rebroadcasts message $m$ to all of its neighbours. If node $S$ receives message $m$ later on it is discarded.

2. For a node $v_i$ when it receives a broadcast message $m$ from a node $v_j$, it performs the following operations:
   
   **If** message $m$ has been transmitted before, then message $m$ is dropped and duplicate messages received later on will be dropped.
   
   **Else if** message $m$ is a new broadcast message, then calculate the additional coverage area $AC$ and initiate a countdown timer. Set it to a random value in the interval $[0, \text{max\_wait\_time}]$.
   
   **Else** update $AC$ and discard message $m$.
   
   **When** the countdown timer expires, if $AC < T_a$, then the rebroadcast of message $m$ is prohibited; otherwise, a rebroadcast of message $m$ occurs.

One problem with this scheme has to do with the complexity in calculating $AC$. This relates to calculating the intersections between circles and it is difficult even when
5.4. NON-NEIGHBOUR KNOWLEDGE METHODS

![Diagram of a 4 node network](image)

**Figure 5.7.** Example using the location-based scheme in a 4 node network.

there are only three circles. An alternative is to use a convex polygon to determine whether to rebroadcast or to drop the message. To illustrate the use of a convex polygon, suppose a node $A$ have received three broadcasts of a message from nodes $B$, $C$ and $D$. As seen in Figure 5.6 a) the additional coverage for $A$ is small or even none when node $A$ is inside the convex polygon formed by the nodes $B$, $C$ and $D$. However, if node $A$ is outside the convex polygon formed by these three nodes, as seen in Figure 5.6 b), node $A$ will provide more additional coverage. Thus, a node is prohibited to rebroadcast a message if it is inside the convex polygon formed by the reception of a broadcast message from the other nodes; otherwise it rebroadcasts the message.

To get a better understanding of this protocol, an example follows. The topology for this example is shown in Figure 5.7. Node $S$ initiates the location-based scheme by a broadcast of message $m$. Nodes $A$, $B$ and $C$ receives the broadcast of message $m$ and each assigns a random delay to its timer. They also calculates the additional coverage and assigns it to $AC$. Lets assume that the additional coverage of these nodes is above the coverage threshold i.e. $AC > T_a$. Lets further assume that the timer at node $A$ expires first. Node $A$ then rebroadcasts message $m$. The nodes $S$ and $B$ receive the broadcast. Node $S$ drops message $m$, since it has already done a broadcast of that message and node $B$ updates $AC$. Assume that the timer at node $C$ expires next. Then, node $C$ rebroadcasts message $m$, which is received by nodes $S$ and $B$. Node $S$ drops message $m$ and node $B$ again updates $AC$. When the timer at node $B$ finally expires, it rebroadcasts message $m$ since $AC \geq T_a$. Nodes $S$, $A$ and $C$ receive the broadcast and drop message $m$. Notice that this example does not use the convex polygon approach.

### 5.4.2 Optimised broadcast protocol

The idea with the optimised broadcast protocol (BPS) [21] is to cover the entire geographic area with as few hexagons as possible, see Figure 5.8 a). The length of the hexagon sides should be determined by the transmission range $d$. When the protocol
is initiated the source node is positioned at the center of a hexagon. In an ideal network, all other rebroadcast nodes are then strategically located at the corners of the hexagons covering the entire geographic area. This is shown in Figure 5.8 b). Since no knowledge about neighbouring nodes is collected by this protocol and as such nodes do not know the positions of the other nodes in its proximity. It adds its own location and the location of the previous broadcast node to the broadcast message. This makes it possible for the node that receives the message to calculate the location of the next strategic locations. To determine the rebroadcast order of nodes close to a strategic location a dynamic delay function is used. This function generates a shorter delay if the node is close to a strategic location and a longer delay if it is further away. To prohibit other nodes to rebroadcast the same message, a fixed threshold $d_{Th}$ is used. A node keeps track of the distance $d_{min}$ to the nearest node that has transmitted the message. It refrains from doing a rebroadcast if $d_{min} \leq d_{Th}$; otherwise a broadcast of the message is commenced.

The broadcast procedure for this protocol is formally derived below.

1. Source node $S$ sets the two location fields $L_1$ and $L_2$ in message $m$ to its own location $(S_X, S_Y)$. Node $S$ then broadcasts message $m$ to all of its neighbours. If node $S$ hears message $m$ later on it is dropped.

2. For a node $v_i$ when it receives a broadcast message $m$ from a node $v_j$, it updates $d_{min}(m)$ if $d_{min}(m) > dist(v_i, v_j)$. Then it performs the following operations:

   If $m$ has been transmitted before or $d_{min}(m) \leq d_{Th}$, then message $m$ is dropped.

   Else if $m$ originates from $S$, then node $v_i$ finds the closest vertex in a hexagon with its center coordinate at $(S_X, S_Y)$ and one of its vertex coordinates at $(S_X + d, S_Y)$. Node $v_i$ then calculates its distance $l$ from the closest vertex and delays the rebroadcast of message $m$ by $D = l/d$. 

76
5.4. NON-NEIGHBOUR KNOWLEDGE METHODS

![Image of a five node network topology](image)

Figure 5.9. Five node network topology used to illustrate how BPS works.

Else node $v_i$ computes the vertex locations based on the locations $L_1$ and $L_2$. It selects the nearest strategic location (i.e. vertex) and calculates the distance $l$ to it. Then the rebroadcast of message $m$ is delayed by $D = l/(20d)$.

When the delay expires. Node $v_i$ again determines if the packet can be discarded (same reasons as above). Otherwise, node $v_i$ updates $L_1$ to the location of node $v_j$ and $L_2$ to its own location, $d_{\text{min}}(m) = 0$ and a rebroadcast of message $m$ occurs.

To get a better understanding how this protocol operates, an example follows. Source node $S$ needs to notify other nodes in its vicinity about an moving object, see Figure 5.9. It initiates a broadcast using the optimised broadcast protocol by transmitting a message $m$. The nodes $A$, $C$ and $D$ receive broadcast message $m$ and each of these nodes decides not to discard message $m$. They notice that it originates from source node $S$, since the location fields $L_1$ and $L_2$ are equal. Each of these nodes computes the strategic locations and finds the one which is closest to their location. Then they set the delay depending on their distance from the closest strategic location. Lets assume that the delay at node $C$ expires first. It determines to rebroadcast message $m$ and assigns $L_1$ to the location of $S$ and $L_2$ to its own location. Then it rebroadcasts message $m$. The nodes $S$, $D$, $A$ and $B$ receive message $m$. Both nodes $S$ and $D$ discards the message, for node $S$ it is because it has retransmitted it before and for node $D$ its distance to node $C$ is below the threshold value (i.e. $d_{\text{min}}(m) < d_{Th}$). Node $A$ reassigns $d_{\text{min}}$ to the node closest of $S$ and $C$ and then continues to countdown its timer. Node $B$ determines not to discard message $m$ and notices that message $m$ is not from the source node. It determines the closest strategical location and calculates its delay. Lets further assume that the timer at node $A$ expires. Node $A$ determines to rebroadcast message $m$ and sets the location fields and then it rebroadcasts message $m$. The nodes $S$, $C$ and $B$ receive the broadcast and both nodes $S$ and $C$ discards...
message \( m \). Node \( B \), on the other hand, updates \( d_{\text{min}} \) to the closest node of \( A \) and \( C \). Then, let’s assume that the timer at node \( B \) expires, it decides to rebroadcast message \( m \). It sets the location fields and rebroadcasts message \( m \). Both node \( A \) and \( C \) receive broadcast message \( m \) and discard message \( m \), since they have already transmitted it.

5.5 Link quality-aided flooding

In this section we present the link quality-aided flooding (LQAF) protocol \([47]\). It divides the received signal strength (RSS) to a number of discrete values, referred to as link quality values. It then uses these values to determine the rebroadcast delay. A low link quality value generates a short delay and a higher link quality value generates a longer delay. Below follows a description of the link quality, then the dynamic delay function and the LQAF protocol.

5.5.1 Link quality

The received signal power at the same distance from the transmitting node varies and depends on small and large scale fading effects \([71]\), see Fig. 5.10. In this algorithm a function is used to convert the received signal power to a link quality value. The link quality is a measure of the reliability of the communication link and links with the same link quality provide the same reliability. A link quality value of zero means that the channel is almost unusable, i.e. it is electrically far away from the broadcast node, and a higher value means a better reliability i.e. it is electrically closer to the broadcast node. By taking into account the reliability of the channel a more robust protocol can be constructed. Such a protocol can utilise the fact that the link quality produce a discrete number of circular regions, in the link quality domain, around the broadcast node. This can be used to find optimal rebroadcast positions in the distance domain.

The link quality \( y_{\text{lq}} \) is obtained by quantizing the received signal power \( r \) according to \( y_{\text{lq}} = Q(r) \), where \( Q(r) \) is a uniform staircase function that takes values from \( Y = \{ k | k = 0, 1, \ldots, S_{\text{lq}} \} \) and

\[
y_{\text{lq}} = k \text{ when } x_k \leq r < x_{k+1},
\]

where \( x_k \) and \( x_{k+1} \) are decision threshold and \( S_{\text{lq}} \in \mathbb{N} \) is a user-defined value that decides the sensitivity of the quantizer, i.e. the number of quantization steps.

5.5.2 Dynamic delay

Since the link quality function has a finite number of discrete steps, the delay among the nodes with the same link quality is the same. Thus, nodes rebroadcast the message at the same time. To avoid this problem a random variable \( X \), that generates
values in the range \([0, 1)\), is added to the cost function. This decreases the possibility that nodes with the same link quality transmit messages at the same time and thereby avoiding collisions, contention and redundancy. The proposed dynamic delay function is defined as

\[ D = \beta(|y_{lq} - T| + \mathcal{X})^\alpha, \]  

(5.2)

where \(\alpha, \beta\) are design constants and \(T \in \mathcal{Y}\) is a value in which the rebroadcast delay is shortest. The main differences between this and other used dynamic delay functions, aside from the use of link quality, are the use of \(T, \alpha, \beta\) and \(\mathcal{X}\) to configure the behaviour of the dynamic delay function. The constant \(\alpha\) and \(\beta\) are used to manipulate the delay range for each link quality value, where \(\alpha\) is used to increment (decrement) the delay for each succeeding link quality value and \(\beta\) is used to decide the slope of the delay function. The random variable is used to decide the distribution of delays for nodes with the same link quality value and should be chosen based on network density. The parameter \(T\) is used to decide the link quality value that have the shortest delay. To not miss any nodes the delay increases symmetrically for link quality values on both sides of the parameter \(T\). The idea to use different values of \(T\) can for example be used by routing protocols to increase the link reliability for the forwarding nodes in their route discovery processes.

### 5.5.3 Protocol

First we describe the protocol and then an example of the protocol is presented. The broadcast procedure of this protocol is formally derived below.

1. Source node \(S\) simply broadcasts message \(m\) to all of its neighbours. If node \(S\) hears message \(m\) again it is discarded.

2. For a node \(v_i\) when it receives a broadcast message \(m\) from a node \(v_j\), it performs the following operations:
Figure 5.11. An example of the LQAF protocol using a five node network.

If message $m$ has been transmitted before or if $c(m) \geq 1$, then message $m$ is discarded, the timer is cancelled and duplicate messages received later on will be dropped.

Else message $m$ is a new broadcast message, then initiate a countdown timer and a counter $c(m) = 1$. Set the timer to the value generated by equation (5.2).

When the countdown timer expires, a rebroadcast of message $m$ occurs.

A node, $S$, wants to reach all nodes in the network with a software update, see Fig. 5.11. Node $S$ then initiates the flooding by broadcasting a message $m$. Both nodes $A$ and $B$ receives the broadcast message $m$ and each node calculates its delay based on the obtained link quality. Since node $B$ has a lower link quality it is assumed that its delay timer expires first and a rebroadcast of message $m$ occurs. Node $S$, $A$, $C$ and $D$ receives the rebroadcast of message $m$ from node $B$. Since node $S$ has done a broadcast of message $m$ before, the message is dropped. Node $A$ on the other hand has received message $m$ before, it cancels its timer and drops the message. Both nodes $C$ and $D$ rebroadcasts the message after their respective timer has expired, since they are not in communication range of each other. Notice that if a node $E$ had been connected to node $A$ only, it will not have received the broadcast of message $m$.

5.6 Received signal strength-aided flooding

In this section, we present the received signal strength-aided flooding (RAF) protocol [50]. The idea of this protocol is to use the received signal strength (RSS) instead of the actual distance to determine the order of rebroadcast nodes. In addition to this mechanism, it also includes an energy aware metric in the dynamic delay function that adjusts the delay depending on the remaining energy of a node. This is a simple and efficient solution, which is preferable in wireless sensor networks.
5.6. RSS-AIDED FLOODING

5.6.1 Received signal strength

The RSS depends on small and large scale fading effects. This means that the RSS varies at the same distance from the transmitting node. This can occur both in different directions and in the same direction but at different time instances [71]. In this paper the RSS in decibel is used as a measure of the electrical transmission distance between a transmitting node and a receiving node. This electrical distance is viewed as the reliability of a communication link and links with the same RSS provide the same reliability. RSS close to the receiver sensitivity means that the link has a low reliability, i.e. the receiving node is far away from the transmitting node. This is in the sense of electrical transmission distance. However, this might not be true in the sense of geographic distance. The other extreme case is when the RSS is close to the transmit power, which means that the link has a high reliability, i.e. the receiving node is close to the transmitting node.

It has been suggested in the literature [35][45] that the RSS can be used to obtain the distance using a propagation model. However, none of these have proposed to use the RSS directly.

5.6.2 Energy-aware dynamic delay

In this work we propose to use the RSS in a dynamic delay function as a measure of the reliability of the communication link. In addition, we combine this with the value of an energy related metric to adjust the delay based on the normalised remaining energy of a node. In equation (5.3) the parameters used are: $S$, the receiver sensitivity in dBm; $r_1$, the received signal strength in dB obtained at the receiving node; $E_l$, the normalised remaining energy at the receiving node; and $E_0$, a threshold that determines when the delay starts to increase due to low remaining energy at a node. The initial delay $D_1$ is obtained from the dynamic delay function and is defined as.

$$D_1 = \begin{cases} 
\max_{\text{wait time}} \frac{|S| - |r_1|}{|S|} & \text{if } E_l \geq E_0 \\
\max_{\text{wait time}} \frac{|S| - |r_1|E_l}{|S|} & \text{if } E_l < E_0 
\end{cases} \tag{5.3}$$

where $\epsilon$ is a configuration constant. This constant is used to determine the slope of the dynamic delay function and an $\epsilon > 1$ spreads the delay over a larger time interval when the received power is close to the transceiver sensitivity and the spreading interval decreases as the RSS becomes better. On the other hand, using an $\epsilon < 1$ the reverse is true. Basically, this dynamic delay function uses the RSS combined with the remaining energy of a node to determine the initial waiting time. A value close to the receiver sensitivity, i.e. a communication link with low reliability, generates a short delay and a RSS value closer to the transmit power generates a higher delay.

\(^1\)It is important to notice that if the $RSS > 0 dB$ the equation does not give us the appropriate results. Thus the transmit power is assumed to be $0 dBm$ or less, which leads to a $RSS \leq 0 dB$. 

81
CHAPTER 5. EFFICIENT FLOODING PROTOCOLS

Figure 5.12. The rebroadcast area for the first redundant broadcast message using the RAF protocol. Thresholds $R_{Th,a}$ for node $a$ and $R_{Th,b}$ for node $b$.

When a redundant broadcast message is received, the initial delay $D_1$ is prolonged according to the following equation

$$D_c = D_{c-1} + \frac{\text{max\_wait\_time}}{c} \quad \forall \ c \geq 2,$$

(5.4)

where $c$ is the number of times the same broadcast message is received. This equation has a bound on the maximum delay that can be obtained. The reason for this is to be able to provide a latency constraint at the network-layer.

5.6.3 Protocol

We begin this subsection with a formal description of the RAF protocol and then we present an example to give an idea how this protocol operates.

The number of received broadcasts of a message $m$ is stored in a counter $c(m)$ and the RSS for these messages are stored in a vector $R = [r_1, r_2, \ldots, r_n]$, where $r_i$ is the RSS for message $0 < i \leq n$. Notice that the number of elements in $R$ increases with each arriving message. The RAF protocol utilises a threshold $R_{Th}$, based on the RSS, to further increase the delay for nodes that receive redundant broadcasts within a geographic area, referred to as the rebroadcast area. A node is always inside the rebroadcast area if $c(m) < 2$; otherwise a redundant message has been received and a node is inside the rebroadcast area if $|r_j| > |R_{Th}| \forall j$, where $0 < j \leq c(m)$; otherwise it is outside the rebroadcast area. The nodes that are outside the rebroadcast area and receive redundant broadcasts are prohibited to rebroadcast the message. This is illustrated in Fig. 5.12.

The algorithm is formally derived below.

82
5.6. RSS-AIDED FLOODING

1. Source node $S$ simply broadcasts message $m$ to all of its neighbours. If node $S$ hears message $m$ again it is discarded.

2. For a node $v_i$ when it receives a broadcast message $m$ from a node $v_j$, it performs the following operations:

   - **If** message $m$ has been transmitted before or $|r_j| \leq |R_{Th}|$, for any $j$, where $0 < j \leq c(m)$, then the timer is cancelled, message $m$ and duplicate messages received later on is discarded.
   
   - **Else If** message $m$ is a new broadcast message, then initiate a counter $c(m) = 1$ and initiate a countdown timer. The countdown timer is set to the value generated by delay function (5.3) and node $v_i$ stores the RSS of message $m$ in $r_{c(m)}$.
   
   - **Else** node $v_i$ updates $c(m) = c(m) + 1$ and stores the RSS of message $m$ in $r_{c(m)}$. It then calculates an extra delay and adds it to the current delay for message $m$, using equation (5.4).

   **When** the countdown timer expires, a rebroadcast of message $m$ occurs.

To get a better understanding of how the proposed protocol works, an example follows. Source node $S$ wants to reach all nodes in the network with a software update, see Fig. 5.13. We assume that the remaining node energy does not influence the rebroadcast delay of the nodes, i.e. $E_i \geq E_0$ in equation (5.3) for all nodes. Node $S$ then initiates the RAF-protocol by a broadcast of message $m$. The nodes $A$, $B$ and $C$ receives the broadcast from node $S$ and initiate the countdown timer according to delay function (5.3). Let's assume that the link $SA$, from node $S$ to $A$, is the most unreliable communication link. In this case, the countdown timer at node $A$ expires first and a rebroadcast of message $m$ occurs. Both node $S$ and $B$ receive the broadcast of message $m$ from node $A$. Node $S$ drops the message, since it has done a broadcast of message $m$ before. The RSS for both links $SB$ and $AB$ have a reliability such that node $B$ is inside the rebroadcast area. Thus, node $B$ prolongs its countdown timer for message $m$, using equation (5.4). Let's further assume that the countdown timer at node $C$ expires after that of node $A$. Thus, node $C$ rebroadcast message $m$ and message $m$ is then received at both node $S$ and $B$. Node $S$ again drops message $m$ and node $B$ stops its countdown timer and refrains from doing a rebroadcast of message $m$. This is because $r_C \leq R_{Th}$ at node $B$, which positions this node outside the rebroadcast area.

Compared to other area-based methods the proposed protocol differs in the following important ways. Firstly, it uses the RSS directly; it helps in deciding which nodes are close to the actual transmission range. This, however, is not always true when using the distance as a measure and that is due to properties of the wireless channel. A benefit of this is that no information is needed to be added to the broadcast message, e.g. distance, which is rather common in this class of flooding protocols. Secondly, it uses a threshold value only for nodes that receive redundant broadcast messages. That is, the first received message is not subjected to a threshold value. This threshold is
mainly used to define a subregion in which nodes are likely to be well positioned to cover a large new area, see Fig. 5.12. Thirdly, it uses an energy-aware dynamic delay function that increases the delay until a rebroadcast may occur based on the remaining node energy.

5.7 Protocol comparison

In this section a comparison of four controlled flooding protocols will be conducted. The flooding protocols compared are the improved flooding with self-pruning (iFSP), optimised broadcast protocol (BPS), link quality-aided flooding (LQAF) and received signal strength-aided flooding (RAF). The first one is a neighbour knowledge method and the other fall into the category of area-based flooding protocols i.e. the non-neighbour knowledge class. The classic flooding protocol is used as a reference. In the comparisons the used metrics are: reachability, the number of nodes that are reached by a broadcast message; the number of rebroadcast nodes; and energy efficiency, which in this case is how long the flooding protocol can operate without reducing its reachability. Below follows a description of the simulation set-up and then the simulation results are presented.

5.7.1 Set-up

During these experiments the network simulator SWANS (scalable wireless ad-hoc network simulator) have been used. This network simulator runs on top of the JIST (Java in simulation time) simulation framework [11]. The simulations are performed in a square region with borders of length 1000 m. See Figure 5.14 for an example placement of 200 nodes. There are $N$ nodes randomly distributed in this region with an average wireless transmission distance $d \approx 110 m$. To more accurately reflect reality the transmission distance is time varying and the variations of the wireless channel is obtained using the free-space propagation model with a Rayleigh fading channel (Configuration option in JIST/SWANS). The simulator takes into account the queuing,
processing delay of messages in the network stack and the transmission time of a message. The transmission bit-rate is set to 256 kbps with a transmission power of 0 dBm, the transceiver sensitivity is $S = -81$ dBm and both the antenna gain and the system loss factor is set to 0 dB.

From an energy consumption perspective in the simulation environment, the sensor node can operate in three different states. These states correspond to sense, transmit and receive states in the complete energy dissipation model, see Chapter 4.4. Thus, during simulation the sensor nodes do not enter the sleep state. In addition to this, the state transitions are neglected. The power consumed for the available operation states is the same as that in Table 4.3. That is the power consumed in the transmit state is 95.3 mW, 98.2 mW in the receive state and 4.8 mW in the sense state.

Below follows parameters for each protocol.

**iFSP:** As mentioned before, iFSP uses neighbour knowledge to make its rebroadcast decisions. The beacon interval to update the neighbour tables is set to 5 s. Since it adds its own neighbour list to each broadcast message the initial size of the MAC frame is 84 bits and each neighbour increases this size by 2 bits, i.e. totally $2|N(v)|$ bits are added where $v$ is the transmitting node. The beacon message size is set to 59 bits and the parameter $\text{max\_wait\_time} = 500$ ms.

**BPS:** BPS adds two location fields to each MAC frame and as such it has a size of 94 bits, where each location field constitutes of 5 bits each. This protocol also uses a threshold value to determine which nodes are allowed to rebroadcast a message and $d_{Th} = 0.5d$. It is stated in [50] that this value of $d_{Th}$ provides a similar reachability to that of the RAF protocol with a threshold $R_{Th} = 0.9|S|$.  

**LQAF:** For this protocol the parameters are set as follows: $\alpha = 2$, provide a higher
probability for nodes to retransmit later in their assigned discrete slot; \( \beta = 2 \), increases the delay for each consecutive discrete slot; \( T = 0 \) is used since as much new geographic area needs to be covered and that no considerations about the reliability is taken into account; and the \( S_{\text{tg}} = 127 \). Lastly, each MAC frame has a size of 84 bits due to that no additional information needs to be stored in the message.

**RAF:** Each MAC frame for this protocol has a size of 84 bits and it is the same reason as that of the LQAF protocol. In addition, the \( \text{max \_ wait \_ time} = 500 \text{ ms} \); \( \epsilon = 2 \), to have a larger distribution of the delay close to \( d \); and \( E_0 = 0 \) for this protocol. The thresholds \( R_{Th} = 0.9|S| \) and \( R_{Th} = 0.95|S| \) are used. The protocol with a \( R_{Th} = 0.9|S| \) is referred to as RAF 0.9 and the other one is referred to as RAF 0.95.

These parameters and their values are used in all simulations if not otherwise stated.

### 5.7.2 Results

In this subsection the results are presented for the contributed protocols and compared to BPS and the iFSP protocol. As a reference classic flooding is used. The comparisons are made in the following order: reachability, number of rebroadcast nodes, and energy efficiency. Reachability is conducted first since the protocols to be compared should be done so on the same basis. This means that the protocols should have a similar reachability so a comparison of the number of rebroadcast nodes can be done fairly.

#### Reachability

In the simulations to obtain the reachability, the number of nodes \( N \) in the network varied from 50 to 700. For values of \( N \) ranging from 50 to 350 nodes the step-size is 25 nodes and for values of \( N \geq 400 \) the step-size is 100 nodes. In the first range 200 simulations and in the second range 50 simulations were conducted and the mean was calculated for each value of \( N \). The calculated mean values are referred to as a simulation run. In total 10 simulations runs were conducted and the mean of these simulation runs were calculated.

Figure 5.15 shows the reachability for the evaluated protocols. As can be seen in this figure the CF protocol has the highest reachability of all protocols during a low network density. Closely following the CF protocol is the RAF 0.9 protocol. The other protocols have similar reachability to each other and both the iFSP and the RAF 0.95 protocol have only marginally higher reachability during low node densities. The worst protocol at low network densities is LQAF, which is slightly worse than BPS. For high network densities the reachability is close to 100% for all compared protocols. The main problem at low network densities for these protocols are the connectivity of the network, which is rather sporadic. As the network density increases so does the connectivity of the network, leading to an increased reachability for all protocols.
5.7. PROTOCOL COMPARISON

**Number of rebroadcasts**

In this simulation the parameter $N$ is varied between 50 and 700. The number of simulations for each value of $N$ and the step-size are the same as those in the previous subsection concerning reachability. The number of simulations runs is also the same. However, in addition to calculating the mean of the simulation runs the variance is also calculated. This provides a good estimation of the errors related to the obtained mean value.

Figure 5.16 shows the percentage of rebroadcast nodes in the sensor area. As can be seen in this figure, the number of rebroadcast nodes rapidly increases as the node density increases. The reason for this is the low connectivity of the network at low node densities, as explained previously. As the connectivity increases the protocols stabilise and the number of rebroadcast nodes starts to degrade. This is due to the increasing number of nodes positioned at the calculated positions in BPS; at the borders of the transmission radius for the RAF and the LQAF protocols; and the increased number of neighbouring nodes that do not reach additional nodes in the iFSP protocol. In Figure 5.16 it can be seen that the LQAF protocol has the lowest number of rebroadcast nodes after it becomes stable. At the highest density the number of rebroadcast nodes for this protocol is below 20%. Closely following the LQAF protocol is the RAF 0.95 protocol, which is roughly 3% worse than the former. The other compared protocol are BPS, which is roughly 10% worse than LQAF; and then iFSP, which is 25% worse than LQAF. It is important to reflect that the LQAF protocol is a special case of the
RAF protocol with a different dynamic delay function and the threshold value of RAF is set to one (i.e. no rebroadcast area exist). As such, the RAF protocol makes a trade-off by increasing the reachability and at the same time it also increases the number of rebroadcast nodes. Furthermore, it is shown in Figure 5.15 that RAF \(0.9\) has a reachability close to that of CF. In addition, from Figure 5.16 it can be seen that the percentage of rebroadcast nodes for RAF \(0.9\) is much lower than that of CF with only a slightly lower reachability.

**Energy efficiency**

In these simulations the number of nodes are constant \(N = 350\), which gives a reachability around 100\% for all protocols. A new broadcast message is transmitted every 5 sec from the base station and the total simulation length is 500 sec. This yields a total of 100 broadcast messages. Each node starts with a energy supply of \(2J\) and a node consumes power at all time. The amount of power consumed depends on its current state i.e. sense, transmit or receive. Once a nodes energy supply is depleted it ceases to function (i.e. it is dead).

Figure 5.17 shows the normalised number of dead nodes at different time instances. It can be seen in this figure that the nodes using the iFSP protocol starts to die first and it occurs at time instance 165 sec. The main reason for this is due to the beacon messages transmitted, which increases the energy consumption due to the extra transmissions and receptions needed. However, this can be adjusted based on the periodicity of the
5.7. PROTOCOL COMPARISON

beacon messages. By increasing the time between beacon messages the time until the first node dies also increases. The reverse is also true. After this protocol, the nodes using the CF protocol start to die at a time instance of 195 sec. The main reason for this is that all nodes rebroadcast the message. At time instance 265 sec the nodes using the RAF 0.9 protocol start to die and at a time instance of 275 sec the nodes using BPS start to die. We expected BPS to last longer compared to the RAF 0.9 protocol. Due to the lower number of rebroadcast nodes as seen in Figure 5.16. From this observation we concludes that the longer MAC-frames transmitted and received for the nodes using BPS has a high influence on the energy consumption. In the RAF 0.95 and the LQAF protocols the nodes start to die at a time instance of 310 sec and 325 sec respectively. This is expected as the number of rebroadcast nodes are lower for the LQAF protocol than that of the RAF 0.95 protocol. As can be seen from these results, the contributed protocols last longer before nodes start to die during periodic broadcasts.

Figure 5.18 shows the reachability of these protocols at different time instances. It can be seen from this figure that the reachability is 100% in the beginning of the simulation. When the number of dead nodes starts to increase, as seen in Figure 5.17, the reachability for the protocols starts to decrease rapidly. The rapid decrease in reachability occurs for nodes using the iFSP protocol at a time instance of 160 sec, for nodes using the CF protocol at 200 sec, for nodes using the RAF 0.9 protocol at time instance 270 sec, for nodes using BPS at 280 sec, for nodes using the RAF 0.95 protocol at 315 sec and for nodes using the LQAF protocol at a time instance of

![Figure 5.17. Normalised number of dead nodes over time.](image-url)
CHAPTER 5. EFFICIENT FLOODING PROTOCOLS

330 sec. The reason for the rapid decrease in reachability is due to the network, which becomes more and more disconnected as time progresses. The nodes that start to die first are those closest to the base station. Once these nodes have died no other nodes can be reached by a broadcast from the base station, which results in a reachability of 0%.

5.8 Conclusion

In this chapter we have presented the background that leads to the idea of controlled flooding protocols and some categorise of flooding protocols. In addition four controlled flooding protocols are compared to each other. These protocols are: improved flooding with self-pruning, optimised broadcast protocol, link quality-aided flooding and received signal strength-aided flooding. The two latter protocols are our contributions to this research field and these two protocols share the same key features, namely:

1. No knowledge about neighbouring nodes is required.
2. No additional overhead in the broadcast message is needed.
3. No location information is needed.

These three key features distinguish the contributed protocols from the neighbour knowledge class and the area-based methods. In this sense, the contributed proto-
cols bear more resemblance to the probabilistic methods. Even though they try to cover as much geographic area as possible with each new broadcast.

From the results the contributed protocols use fewer rebroadcast nodes than that of BPS and the iFSP protocols with a similar reachability. All the compared protocols, except CF, lower the number of rebroadcast nodes as the density in the network increases. The results further show that the lifetime of the sensor nodes using the contributed protocols are longer than that of the other protocols. This is due to the fewer number of rebroadcast nodes in the network. In addition, adding information to the broadcast message has a high impact on the lifetime of the sensor nodes and as such should be avoided.

The features and the results of the contributed protocols makes them good candidates to be used as building blocks for more advance protocols like routing and cluster construction protocols.
Chapter 6

Conclusion

In this thesis we have presented the research in the areas of energy dissipation models and efficient flooding protocols for wireless sensor networks. Energy dissipation models are used for evaluation of different aspects of a sensor node. Efficient flooding is used to alleviate the problem that can occur due to a broadcast storm.

In the area of energy dissipation models our contribution in this thesis is the complete energy dissipation model. It alleviates some of the problems that other energy dissipation models have, like not including all subsystems and assuming that state transitions are instantaneous. The contributed model is based on the basic sensor node architecture and as such takes into account all of its subsystems. This model defines four operations states and the corresponding state transitions. These operation states provide a good accuracy in terms of energy dissipation, while the energy overhead that occur during a state transition can be used for accurate scheduling decisions. In addition to the energy dissipation model, the measurements for the operation states and state transitions are also provided for this energy model. The evaluation of the model shows promising results in which the estimated results has a deviation of less than 1% to those of the measured values.

In the research area of efficient flooding, two protocols are contributed and these are link quality-aided flooding (LQAF) and received signal strength-aided flooding (RAF). The former protocol can be seen as a special case of the latter protocol with its own unique dynamic delay function. The latter protocol is developed to be more adaptable than the former, in terms of both reachability and number of rebroadcast nodes. Since the two protocol share the same basic idea they also share the same key features: no knowledge about neighbouring nodes, no additional overhead in the broadcast message and no location information are required. From the results obtained, it is shown that these two protocols perform much better than the others. The number of rebroadcast nodes are much lower with similar reachability and in terms of energy efficiency the nodes using the contributed protocols last longer than the others. The results and the key features of these two protocols make them good candidates to be used as building blocks in more advanced network protocols.
6.1 Future work

Below follows some ideas that need to be investigated and developed further. This includes an improved energy dissipation model and investigations to improve the efficient flooding protocol. In addition to these suggestions, a research direction is briefly presented in which our contribution in Chapter 4 is going to be used for evaluation and the contributions in Chapter 5 are used as building blocks to construct a more advanced network protocol.

6.1.1 Energy dissipation model

Below follow some suggestions for future work related to Chapter 4.

Extended energy model: Since the complete energy dissipation model only includes the basic sensor node architecture. An extended model should be developed that incorporates additional subsystems. In addition, a framework that allows this extended energy model to be adapted for different sensor nodes should be further investigated.

Measurement: It relates to gathering measurements from other sensor nodes and to evaluate the complete energy dissipation model using more scenarios based on the obtained measurements. Furthermore, the accuracy and the variations of the measured results and the evaluations should be included.

6.1.2 Efficient flooding

Suggestions for future work related to Chapter 5 is described below.

Delay distribution: Investigate how different delay functions influences the retransmission behaviour of the RAF protocol. It also includes how to obtain a good time distribution for the retransmission delay based on the average density of the sensor field. This time distribution is then compared to that based on the transmitting node degree.

Adjusting threshold: Investigate how to adjust the threshold of the RAF protocol based on global density of the sensor field. Then extend this idea to use local density (i.e. density of the neighbours) instead of global density. It is likely that this extension would make the protocol even more adaptable to the constant changes in the topology of the network, especially if the local density fluctuates heavily.
6.1.3 Direction

The future direction for the work presented in this thesis is to develop a cluster construction and management algorithm for use in wireless sensor networks. The efficient flooding protocol, RAF, is going to be further improved to be able to act as a building block in the cluster construction algorithm. Basically, the cluster construction algorithm chooses as its clusterheads the rebroadcast nodes in the RAF-protocol. The other nodes, based on their received signal strength, decide which cluster to join. This constructs a communication path between neighbouring clusterheads. As such, communication within a cluster can use a lower transmission power.

The clusterheads, when established, obtain their location based on a location estimation algorithm. From this knowledge the clusterheads can estimate their coverage areas using triangulation between the clusterheads. As this is accomplished, a sleep scheduling algorithm can make decisions, based on the clusterheads knowledge, concerning sleep times for nodes in its cluster.

To evaluate the energy consumption and the scheduling decisions of this algorithm the complete energy model and its related measurements are going to be used.
References


[8] Sunspot: Sun small programmable object technology.


REFERENCES


REFERENCES


REFERENCES


REFERENCES


