Algorithms and Protocols Enhancing Mobility Support for Wireless Sensor Networks Based on Bluetooth and Zigbee

Javier García Castaño

September 2006
Abstract

Mobile communication systems are experiencing a huge growth. While traditional communication paradigms deal with fixed networks, mobility raises a new set of questions, techniques, and solutions. This work focuses on wireless sensor networks (WSNs) where each node is a mobile device. The main objectives of this thesis have been to develop algorithms and protocols enabling WSNs with a special interest in overcoming mobility support limitations of standards such as Bluetooth and Zigbee. The contributions of this work may be divided in four major parts related to mobility support. The first part describes the implementation of local positioning services in Bluetooth since local positioning is not supported in Bluetooth v1.1. The obtained results are used in later implemented handover algorithms in terms of deciding when to perform the handover. Moreover local positioning information may be used in further developed routing protocols. The second part deals with handover as a solution to overcome the getting out of range problem. Algorithms for handover have been implemented enabling mobility in Bluetooth infrastructure networks. The principal achievement in this part is the significant reduction of handover latency since sensor cost and quality of service are directly affected by this parameter. The third part solves the routing problems originated with handovers. The main contribution of this part is the impact of the Bluetooth scatternet formation and routing protocols, for multi-hop data transmissions, in the system quality of service. The final part is a comparison between Bluetooth and Zigbee in terms of mobility support. The main outcome of this comparison resides on the conclusions, which can be used as a technology election guide.

The main scientific contribution relies on the implementation of a mobile WSN with Bluetooth v1.1 inside the scope of the ”Multi Monitoring Medical Chip (M3C) for Home-care Applications” European Union project (Sixth Framework Program (FP6) Reference: 508291) offering multi-hop routing support and improvements in handover latencies with aid of local positioning services.
Acknowledgments

It has been a good, long journey, full of constructive experiences and invaluable development of both technical and personal skills. This is how I describe my PhD studies during the last five years in Sweden. I would like to express my gratitude to Hans Berggren, the former headmaster of the Electronics Department and Professor Ylva Bäcklund, headmaster of the Department of Computer Science and Electronics at Mälardalen University for inviting me to start this journey.

I would like to take this opportunity to also thank everyone that have contributed to the realization of this work, whether if it has been through technical or emotional support. I am very grateful to my supervisor Mikael Ekström who has always trusted in my judgement and supported me. Thank you for the proof reading of this book, for always being available and all the discussions as partners and as friends.

Thanks to all my colleagues at the department, especially to my friend Jens Lönnblad for all the outdoor activities (how to survive in the snow) and support during all these years. I am also very thankful to Fredrik Granstedt who was the perfect room mate.

I am also very grateful to Anders Martinsen from Mälardalen University, Christopher Gerdtman from Motion Control AB, Mats Wahlén from VG power AB and Jimmy Kjellsson from ABB Corporate Research for sharing knowledge and teaching me how to cooperate and work with companies. Many thanks to the Sixth European Union Research Program, Vinnova and Stiftelsen för Kunskap och Kompetensutveckling; which have economically supported the costs of this work.

Thank you to all good friends in Spain and in Sweden that have never failed and have showed interest in the development of this work.

Many thanks to PhD Ali Fard for the hard scientific proof reading of this book and for being an excellent friend. Thank your for all the stimulating and constructive conversations. Thank you for being a motivation source and for all the funny extra activities including trips, sports and ’’trabajos a casa’’.

Tusen tack Sverige (thank you Sweden), the country that has welcomed and helped me to achieve the results of this work and most important to grow as a person.

And of course I must thank my whole family, especially my parents Andrés and Rosi, my aunt Carmen, my uncle Inocente and my brother Andrés, for the unconditional love, support and for being the cornerstone of my life. Thank you mother for teaching me the ”keep always improving” spirit. Thank you father for teaching me how to enjoy the well-done work. Thank you brother for being the best friend one can imagine and for your technical contribution to this thesis.

An engineering day in June 2006.
/Javier
Agradecimientos

Muchísimas gracias a toda mi familia, en especial mis padres Andrés y Rosi, mis tíos Inocente y Carmen, y mi hermano Andrés. Gracias por el amor y el apoyo incondicional y por ser la piedra angular de mi vida. Muchas gracias madre por enseñarme a siempre querer mejorar. Gracias padre por enseñarme a encontrar la satisfacción en el trabajo bien hecho. Gracias hermano por ser el mejor amigo que uno puede imaginar y por tu aportación a esta tesis.

Un día de ingeniería en Junio de 2006
/Javier
<table>
<thead>
<tr>
<th>Acronyms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi/4$ DQPSK</td>
<td>$\pi/4$ Differential Quadrature Phase Shift Keying</td>
</tr>
<tr>
<td>8 DPSK</td>
<td>Eight Differential Phase Shift Keying</td>
</tr>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>ACL</td>
<td>Asynchronous Connection Less</td>
</tr>
<tr>
<td>ADC</td>
<td>Analog to Digital Converter</td>
</tr>
<tr>
<td>A-FHSS</td>
<td>Adaptive Frequency Hopping Spread Spectrum</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad hoc on Demand Distance Vector</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>APP</td>
<td>Application Layer</td>
</tr>
<tr>
<td>ARQ</td>
<td>Automatic Repeat Request</td>
</tr>
<tr>
<td>ATMS</td>
<td>Authenticated Tracking and Monitoring System</td>
</tr>
<tr>
<td>BB</td>
<td>Baseband</td>
</tr>
<tr>
<td>BC2</td>
<td>Bluecore2-External</td>
</tr>
<tr>
<td>BDA</td>
<td>Bluetooth Device Address</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>BNEP</td>
<td>Bluetooth Network Encapsulation Protocol</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CAN</td>
<td>Control Area Network</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>Cell-ID</td>
<td>Cell Identification</td>
</tr>
<tr>
<td>CFP</td>
<td>Contention Free Period</td>
</tr>
<tr>
<td>CLKN</td>
<td>Native Clock</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Metal-Oxide-Semiconductor</td>
</tr>
<tr>
<td>CSMA-CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>CSR</td>
<td>Cambridge Silicon Radio</td>
</tr>
<tr>
<td>dBm</td>
<td>Decibels related to 1 milliwatt</td>
</tr>
<tr>
<td>DECT</td>
<td>Digital Enhanced Cordless Telecommunications</td>
</tr>
<tr>
<td>DH1-5</td>
<td>Data High-rate (1 to 5 slots)</td>
</tr>
<tr>
<td>DM1-5</td>
<td>Data Medium-rate (1 to 5 slots)</td>
</tr>
<tr>
<td>DM</td>
<td>Device Manager</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct Sequence Spread Spectrum</td>
</tr>
<tr>
<td>ED</td>
<td>Energy Detection</td>
</tr>
<tr>
<td>EDR</td>
<td>Enhanced Data Rate</td>
</tr>
<tr>
<td>EID</td>
<td>Extended IDentifier</td>
</tr>
<tr>
<td>EIR</td>
<td>Extended Inquiry Response</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FFD</td>
<td>Full functional device</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>FHS</td>
<td>Frequency Hop Synchronization</td>
</tr>
<tr>
<td>FHSS</td>
<td>Frequency Hopping Spread Spectrum</td>
</tr>
<tr>
<td>FM</td>
<td>Frequency Modulation</td>
</tr>
<tr>
<td>FP6</td>
<td>Sixth Framework Program</td>
</tr>
<tr>
<td>GAP</td>
<td>Generic Access Profile</td>
</tr>
<tr>
<td>GFSK</td>
<td>Gaussian Frequency Shift Keying</td>
</tr>
<tr>
<td>GPIB</td>
<td>General Purpose Interface Bus</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GR</td>
<td>Golden Range</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>GTS</td>
<td>Guaranteed Time Slot</td>
</tr>
<tr>
<td>HCI</td>
<td>Host Controller Interface</td>
</tr>
<tr>
<td>HVAC</td>
<td>Heating, Ventilating and Air Conditioning</td>
</tr>
<tr>
<td>ID</td>
<td>Identifier</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial, Scientific and Medical</td>
</tr>
<tr>
<td>ISO</td>
<td>International Standard Organization</td>
</tr>
<tr>
<td>KB</td>
<td>Kilo Byte</td>
</tr>
<tr>
<td>Kbps</td>
<td>Kilo Bits Per Second</td>
</tr>
<tr>
<td>L2CAP</td>
<td>Logical Link Control and Adaptation Protocol</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LM</td>
<td>Link Manager</td>
</tr>
<tr>
<td>LQ</td>
<td>Link Quality</td>
</tr>
<tr>
<td>LQI</td>
<td>Link Quality Indicator</td>
</tr>
<tr>
<td>M3C</td>
<td>Multi Monitoring Medical Chip</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Layer</td>
</tr>
<tr>
<td>MAHO</td>
<td>Mobile Assisted Handover</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Ad-hoc Networks</td>
</tr>
<tr>
<td>MCHO</td>
<td>Mobile Controlled Handoff</td>
</tr>
<tr>
<td>MCU</td>
<td>Micro-Controller</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro Electro Mechanical System</td>
</tr>
<tr>
<td>MIT</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td>MSC</td>
<td>Mobile Switch Center</td>
</tr>
<tr>
<td>NCAP</td>
<td>Network Capable Application Processor</td>
</tr>
<tr>
<td>NCHO</td>
<td>Network Controlled Handover</td>
</tr>
<tr>
<td>NS</td>
<td>Network Simulator</td>
</tr>
<tr>
<td>NWK</td>
<td>Network Layer</td>
</tr>
<tr>
<td>O-QPSK</td>
<td>Offset Quadrature Phase Shift Keying</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnect</td>
</tr>
<tr>
<td>PA</td>
<td>Power Amplifier</td>
</tr>
<tr>
<td>PAN</td>
<td>Personal Area Network</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical Layer</td>
</tr>
<tr>
<td>PSKEY</td>
<td>Persistent Store Key</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RDP</td>
<td>Route Discovery Packet</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RFCOMM</td>
<td>Radio Frequency Communications</td>
</tr>
<tr>
<td>RFD</td>
<td>Reduced Function Device</td>
</tr>
<tr>
<td>RREP</td>
<td>Route Reply Packet</td>
</tr>
<tr>
<td>RREQ</td>
<td>Route Request Packet</td>
</tr>
<tr>
<td>RRP</td>
<td>Route Reply Packet</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Indicator</td>
</tr>
<tr>
<td>SCO</td>
<td>Synchronous Connection Oriented</td>
</tr>
<tr>
<td>SDP</td>
<td>Service Discovery Protocol</td>
</tr>
<tr>
<td>SIG</td>
<td>Special Interest Group</td>
</tr>
<tr>
<td>SoC</td>
<td>System on a Chip</td>
</tr>
<tr>
<td>SPP</td>
<td>Serial Port Profile</td>
</tr>
<tr>
<td>Std</td>
<td>Standard</td>
</tr>
<tr>
<td>TDD</td>
<td>Time Division Duplex</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TEDS</td>
<td>Transducer Electronic Data-sheet</td>
</tr>
<tr>
<td>TI</td>
<td>Texas Instruments</td>
</tr>
<tr>
<td>TPC</td>
<td>Transmit Power Control</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver Transmitter</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>VCO</td>
<td>Voltage Controlled Oscillator</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WPAN</td>
<td>Wireless Personal Area Networks</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
<tr>
<td>ZDO</td>
<td>Zigbee Device Object</td>
</tr>
</tbody>
</table>
# Contents

1 Motivation  
1.1 Scope of this Work ........................................ 2  
1.2 Scientific Contributions .................................. 3  
1.3 Other Related Publications ................................. 5  
1.4 Author’s Contributions in the Included Publications . . 6  
1.5 Thesis Organization ........................................ 7  

2 Introduction  
2.1 Wireless Sensor Networks (WSNs) .......................... 9  
2.1.1 WSNs Applications ...................................... 10  
2.1.2 WSNs Design Challenges ............................... 11  
2.1.3 WSNs Projects, Standards and Platforms ............... 13  
2.2 Mobile Wireless Communications .......................... 17  
2.2.1 Mobile Wireless Communication Applications ......... 18  
2.2.2 Mobile Wireless Communication Design Challenges . . 19  
2.2.3 Mobile WSNs ............................................. 21  
2.3 Bluetooth Basics ............................................ 23  
2.3.1 Bluetooth Overview and Features ........................ 24  
2.3.2 Bluetooth Protocol Stack ................................ 25  
2.3.3 Bluetooth Network Topologies ........................... 27  
2.3.4 Bluetooth Voice and Data Packets ........................ 31  
2.3.5 Mobile WSN Support in Bluetooth ...................... 31  
2.4 Zigbee Basics ............................................... 34  
2.4.1 Zigbee Overview & Features ............................. 34  
2.4.2 Zigbee Protocol Stack .................................... 34  
2.4.3 IEEE Std 802.15.4 Overview ............................ 40  
2.4.4 Zigbee Network Topologies ............................... 41  
2.4.5 Mobile WSN Support in Zigbee .......................... 44  
2.5 Chapter Summary ............................................ 45
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Local Positioning with Bluetooth</td>
<td>47</td>
</tr>
<tr>
<td>3.1</td>
<td>Problem Definition</td>
<td>48</td>
</tr>
<tr>
<td>3.2</td>
<td>Goal: Provide Local Positioning Services to Bluetooth</td>
<td>48</td>
</tr>
<tr>
<td>3.3</td>
<td>Relevant Existing Positioning Techniques</td>
<td>49</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Carrier Phase</td>
<td>49</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Radio Signal Strength</td>
<td>49</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Time of Arrival (TOA)</td>
<td>49</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Angle of Arrival (AOA)</td>
<td>49</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Cell Identification (Cell-ID)</td>
<td>50</td>
</tr>
<tr>
<td>3.4</td>
<td>Distance Estimation with Bluetooth</td>
<td>50</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Transmit Power Control (TPC)</td>
<td>51</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Received Signal Strength Indicator (RSSI)</td>
<td>51</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Golden Receiver Range (GR)</td>
<td>51</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Transmitted Power</td>
<td>52</td>
</tr>
<tr>
<td>3.4.5</td>
<td>Link Quality (LQ)</td>
<td>53</td>
</tr>
<tr>
<td>3.5</td>
<td>Bluetooth Mobile Radio Propagation Model</td>
<td>53</td>
</tr>
<tr>
<td>3.6</td>
<td>CSR Bluetooth Implementation</td>
<td>55</td>
</tr>
<tr>
<td>3.6.1</td>
<td>CSR TPC</td>
<td>55</td>
</tr>
<tr>
<td>3.6.2</td>
<td>CSR RSSI</td>
<td>55</td>
</tr>
<tr>
<td>3.6.3</td>
<td>CSR GR</td>
<td>56</td>
</tr>
<tr>
<td>3.6.4</td>
<td>CSR LQ</td>
<td>56</td>
</tr>
<tr>
<td>3.7</td>
<td>Distance Estimation Algorithm</td>
<td>56</td>
</tr>
<tr>
<td>3.8</td>
<td>Triangulation Algorithm</td>
<td>57</td>
</tr>
<tr>
<td>3.8.1</td>
<td>Self &amp; Remote Positioning Topologies</td>
<td>59</td>
</tr>
<tr>
<td>3.8.2</td>
<td>Bluetooth Topology</td>
<td>60</td>
</tr>
<tr>
<td>3.9</td>
<td>Measurements</td>
<td>61</td>
</tr>
<tr>
<td>3.9.1</td>
<td>Measurements Setup</td>
<td>62</td>
</tr>
<tr>
<td>3.9.2</td>
<td>Measurements Procedures</td>
<td>62</td>
</tr>
<tr>
<td>3.9.3</td>
<td>RSSI Measurements</td>
<td>63</td>
</tr>
<tr>
<td>3.9.4</td>
<td>LQ Measurements</td>
<td>64</td>
</tr>
<tr>
<td>3.9.5</td>
<td>Transmitted Power Measurements</td>
<td>64</td>
</tr>
<tr>
<td>3.9.6</td>
<td>Triangulation Measurements</td>
<td>65</td>
</tr>
<tr>
<td>3.9.7</td>
<td>Distance Estimation Accuracy</td>
<td>66</td>
</tr>
<tr>
<td>3.10</td>
<td>Chapter Summary</td>
<td>67</td>
</tr>
<tr>
<td>4</td>
<td>Handover with Bluetooth</td>
<td>69</td>
</tr>
<tr>
<td>4.1</td>
<td>Problem Definition</td>
<td>70</td>
</tr>
<tr>
<td>4.2</td>
<td>Goal: Provide Handover Capabilities to Bluetooth</td>
<td>70</td>
</tr>
<tr>
<td>4.3</td>
<td>Existing Handover Strategies, Types and Algorithms</td>
<td>71</td>
</tr>
<tr>
<td>4.4</td>
<td>Handover Support in Bluetooth</td>
<td>73</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Bluetooth Parameters for Handover Algorithm Implementation</td>
<td>74</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Time Analysis of Bluetooth Connections</td>
<td>74</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Bluetooth Scatternets and Handovers</td>
<td>75</td>
</tr>
</tbody>
</table>
### 4.4.4 ACL and SCO Links Interaction with Inquiry and Page

4.5 Design of Bluetooth Hard Handover Algorithms

#### 4.5.1 Algorithm for Hard Handovers with Mobile Master Sensor

#### 4.5.2 Algorithm for Hard Handovers with Mobile Slave Sensor

4.6 Design of Bluetooth Soft Handover Algorithms

#### 4.6.1 Algorithm for Soft Handovers with Mobile Master Sensor

#### 4.6.2 Algorithm for Soft Handovers with Mobile Slave Sensor

4.7 Resume of Bluetooth Handover Algorithms

4.8 Measurements

#### 4.8.1 Measurements Setup

#### 4.8.2 Measurements Procedures

#### 4.8.3 Functional Measurements

#### 4.8.4 Hard Handover Measurements

#### 4.8.5 Soft Handover Measurements

#### 4.8.6 Results Analysis

4.9 Chapter Summary

### 5 Ad-hoc Routing with Bluetooth

5.1 Problem Definition

5.2 Goal: Provide Ad-hoc Routing Capabilities to Bluetooth

5.3 Existing Ad-hoc Routing Protocols

#### 5.3.1 What is routing?

#### 5.3.2 Table-Driven Routing Protocols

#### 5.3.3 On-Demand Routing Protocols

5.4 Existing Ad-hoc Routing Protocols for Bluetooth

#### 5.4.1 On-demand Scatternet Formation

#### 5.4.2 Dynamic Source Routing (DSR)

#### 5.4.3 BlueStars

#### 5.4.4 BlueTree

#### 5.4.5 BlueRing

5.5 Routing Support in Bluetooth

5.6 Bluetooth Ad-hoc Routing Protocol

#### 5.6.1 Flooding-based Route Discovery

#### 5.6.2 Backward Scatternet Formation

#### 5.6.3 Implementation

5.7 Resume of the Developed Ad-hoc Routing Protocol

5.8 Measurements

#### 5.8.1 Measurements Setup

#### 5.8.2 Measurements Procedures

#### 5.8.3 Inquiry Time Measurements

#### 5.8.4 Page Time Measurements

#### 5.8.5 Route Establishment Time

#### 5.8.6 Throughput
5.8.7 Results Analysis ........................................ 119
5.9 Chapter Summary ........................................ 121

6 Bluetooth and Zigbee Mobility Comparison 123
6.1 Goal: Comparison from a Mobility Point of View ............... 124
6.2 Main Differences within Bluetooth and Zigbee ................. 124
6.3 Mobile WSN support in Zigbee ................................ 127
  6.3.1 Local Positioning in Zigbee ............................ 127
  6.3.2 Handover in Zigbee .................................... 128
  6.3.3 Ad-hoc Routing in Zigbee ............................ 129
6.4 Distance Estimation with Zigbee ................................ 131
  6.4.1 Zigbee Mobile Radio Propagation Model .................. 131
  6.4.2 Algorithm for Distance Estimation with Zigbee ............ 132
  6.4.3 Measurements ........................................ 133
  6.4.4 Accuracy Study ..................................... 133
  6.4.5 Comparison with Bluetooth ............................ 134
6.5 Handover for Zigbee .................................... 134
  6.5.1 Algorithm for Inter-PAN Handover ....................... 135
  6.5.2 Algorithm for Intra-PAN Handover ....................... 136
  6.5.3 Measurements ........................................ 136
  6.5.4 Handover Latency Study .............................. 141
  6.5.5 Comparison with Bluetooth ............................ 141
6.6 Performance of Zigbee Routing Protocol ........................ 142
  6.6.1 Measurements ........................................ 142
  6.6.2 Comparison with Bluetooth ............................ 144
6.7 Chapter Summary ........................................ 144

7 Concluding Remarks 147
7.1 Future Research & Improvements ............................... 150
Chapter 1

Motivation

The demands for public health services are developing rapidly, the challenge is to raise and maintain the present level of health care provision without ending up in an uncontrolled cost explosion [1]. Moreover process automation and manufacturing systems also demand cost reductions, more flexible and more reliable systems. These demands require new approaches rather than traditional way of thinking. One approach widely accepted in recent years is the usage of wireless communications for both biomedical (in particular wireless patient monitoring) and industrial applications. The wireless communication approach assures flexibility, mobility, quality improvements and costs reductions on patient care, system installation, maintenance, amount of cable and fault isolation time [2]. In particular the usage of wireless sensors have become a very popular research and development topic during the last few years. Wireless sensing applications rely on proprietary radio solutions and standards (Std’s) such as Wireless Local Area Networks (WLAN) from the Institute of Electrical & Electronics Engineers (IEEE) Std 802.11 [3], Zigbee [4] and Bluetooth [5] among others. The paradigm of wireless sensing is to provide a reliable, flexible, low cost solution that is easy to deploy for the final user. The critical key parameters that define the quality of a wireless sensor are weight, featured size, battery life, mobility support, data-rate, range and inter-operability [6].

This work focuses on the implementation of mobile wireless sensors with two short range radio standards, Bluetooth v1.1 and Zigbee v1.1. According to Karl in ref. [7], the design of sensor functional blocks has a direct impact in sensor weight, size, features and power consumption. Most of systems lead to single-chip solutions [8] reducing power consumption and size. Embedding these sensors with short range radio standards provides inter-operability and cost reductions for the final user [6]. However, this fact often presents new challenges for the designer who has to adapt already defined standards to provide particular customer solutions. Although most of the required customer applications (cable replacement) are covered
by the primary standard specifications, there are other features like mobility which are barely described and sometimes not supported by the standards. In this thesis the main requirement for the performed work in the M3C project is to provide mobility support for the WSN. There exist specific mobility intended technologies such as Digital Enhanced Cordless Telecommunications (DECT) and Global System for Mobile Communications (GSM) [9]; the higher power consumption required by these standards makes their use in WSN an unwise solution. Therefore Bluetooth and Zigbee which are originally not intended for mobile applications are utilized in this thesis. The research efforts then aim to provide new algorithms and protocols based on elementary standard functionality to overcome mobility support lack [10] in those standards. Moreover the solution must still be compatible with standards to maintain inter-operability. These aspects motivate further research and investigations about implementation of mobility to provide ubiquitous functionality for the final user.

1.1 Scope of this Work

This thesis includes extensive analysis and investigations of support and limitations for mobility management in short range radio standards such as Bluetooth and Zigbee. The work investigates answers to the following questions: what happens when a mobile sensor gets out of range from an access point (AP) or another relaying node? How can the problem be solved using technologies which are not originally intended to support mobility? The answer to the first question is not specific to any technology or communication standard and includes the investigations of positioning, handover and routing techniques. Positioning is used to find the distance between two transceivers and also the geographical position of a sensor. This information aids handover algorithms telling "when" the handover shall be performed. Once the sensor has performed handover there is an important issue that must be solved; how does the system reallocate data streams in order to reach the new sensor location? Routing is then needed in order to reach the desired sensor node. The second question is answered through the different investigations and implementations with Bluetooth and Zigbee. The main goal is to adapt these two technologies using standard functionality to implement mobility services which were not firstly supported. The work includes studies, different implementations, results and suggestions for further improvements. Moreover the performed work covers solutions for infrastructure and ad-hoc WSNs. Impacts of the mobility services in terms of quality of service (QoS), size and sensor cost are then investigated and exposed. The obtained results enable telemetry and sensing applications with mobile sensors (reducing mobility negative impacts).
1.2 Scientific Contributions

New results in this thesis are as follows:

- In Chapter 2, an overview of the challenges associated with the design of mobile WSNs is provided. The overview covers different existing sensor platforms including the sensors developed during this work. The two different wireless sensor architectures developed in this work are described covering one-CPU and stand-alone solutions.

Parts of the material in this chapter have been presented at the following conferences:


- In Chapter 3, local positioning support in Bluetooth is investigated. Investigations conduce to the implementation of a 2-dimensional positioning system using a distance estimation model derived from Friis free space equation. Thereby distances between two Bluetooth transceivers are estimated from signal level measurements. Received Signal Strength Indicator (RSSI) and transmitted power are investigated and tested. Cambridge Silicon Radio (CSR) Bluecore2-external (BC2) modules are used to implement self local positioning systems. The main contribution of this part is the implementation of a Bluetooth local positioning system based on both RSSI and transmitted power readings. Thereby covering all Bluetooth device classes.

The local positioning system with Bluetooth is presented at:

• In Chapter 4, handover at application level with Bluetooth is investigated and implemented. Implementations for infrastructure networks are presented. Distance estimation algorithms from Chapter 3 are then used to execute handover with CSR Bluetooth modules and Linux based solutions. Mobility through APs has been achieved with hard and soft handovers. The main contribution of this chapter is the theoretical investigation of several handover strategies and types applied to Bluetooth, highlighting the benefits and drawbacks of them in terms of system QoS and sensor memory requirements. Moreover the work provides experimental results of the proposed algorithms, a task that is commonly replaced by simulations. Deployed soft handover algorithms ensure that the sensor is always connected to one AP thus reducing handover latency to few hundredths of milliseconds.

Parts of the material in this chapter are available at the proceedings of the following conferences and M3C project report:


• In Chapter 5, ad-hoc routing protocols for WSNs are investigated. The investigations lead to a combination of two previous works, the DSR protocol and the on-demand scatternet formation protocol. The combination of these two works results in the design an implementation of a new routing protocol for multi-hop Bluetooth-based WSNs. The particular Bluetooth baseband implementation sets a relationship between scatternet formation and route discovery. Thereby the developed ad-hoc routing protocol combines ad-hoc scatternet formation and on-demand route discovery. The principal outcomes of this part include the description of an new on-demand ad-hoc routing protocol with the characterization through experimentation of routing effects in terms of route formation time and route throughput.

Parts of the material in this chapter are available at the proceedings of the following conferences and M3C project report:
1.3. OTHER RELATED PUBLICATIONS


- In Chapter 6, a comparison between Bluetooth and Zigbee from a mobility point of view is provided. The Bluetooth performing results are based on the previous mentioned chapters. Zigbee support for mobile applications is investigated and tested to provide a qualitative and quantitative comparison with previous Bluetooth results. The main contribution of this part is the comparison result and the novel Zigbee handover algorithms.

Parts of the work of the chapter is available in the following M3C project report:


1.3 Other Related Publications

- Development of IEEE Std 488.1/2 commonly known as General Purpose Interface Bus (GPIB) cable replacement applications to provide automatic wireless test systems. A digital interface between Bluetooth and GPIB was developed.

This work was presented at:

1.4 Author’s Contributions in the Included Publications

**Paper I:** Partly involved in the software system design and development. Description for the implementation of the Host Controller Interface (HCI) and Logical Link and Control Adaptation Protocol (L2CAP) layers.

**Paper II:** Partly involved in the software system design and development. Description for the implementation of the HCI and L2CAP layers. Communication session ideas, minor parts of the manuscript and combined presentation of the paper.

**Papers III:** Complete hardware and software architecture design and implementation. Major part of the manuscript and presentation of the paper.

**Papers IV:** Complete software architecture design and integration, design and integration of the digital hardware part for single-chip solution. Crucial parts of the manuscript.

**Paper V:** Complete hardware and software architecture design, integration and test of the distance estimation algorithm for both RSSI and transmitted power methods. Major part of the manuscript and presentation of the paper.

**Paper VI:** Complete hardware and software architecture design. Implementation and test of hard and soft handovers based on CSR BC2. Major part of the manuscript and combined presentation of the paper.

**Paper VII:** Complete concept design. Major part of the manuscript and presentation of the paper.

**Paper VIII:** Complete hardware and software architecture design and integration, modifications of Bluetooth stack and performance experiments. Major part of the manuscript and presentation of the paper.

**Paper IX:** Complete hardware and software architecture design, implementation and test of the scatternet route based on CSR BC2. Major part of the manuscript and combined presentation of the paper.

**Paper X:** Complete hardware and software architecture design and integration, performance experiments. Major part of the manuscript and presentation of the paper.

**Paper XI:** Complete software architecture design. Major part of the manuscript
and presentation of the paper.

**Paper XII:** Complete hardware and software architecture design. Full Zigbee investigations and designs. Results analysis. Major part of the manuscript.

### 1.5 Thesis Organization

The rest of this thesis is organized as follows. An introduction to mobility issues in WSNs is presented in Chapter 2, covering a survey of mobility aspects in WSNs together with a survey of the two utilized technologies (Bluetooth and Zigbee), moreover descriptions of the employed sensors are provided. Chapter 3, focuses on studies and design of local positioning systems with Bluetooth. An overview of different positioning techniques and algorithms is first presented. The implementation issues with Bluetooth are described and results of a functional system are reported.

In Chapter 4, several approaches for handover with Bluetooth are described. A theoretical overview of different solutions for Bluetooth is given based on standard and non standard functionality. The theoretical results are deployed in Personal Computer (PC) based solutions. The performed experiments with focus on the handover latency and QoS confirming the theoretical studies are reported. In Chapter 5 the Bluetooth ad-hoc routing protocol is presented. A survey of routing techniques for ad-hoc wireless networks is reported. Route formation theoretical studies for on-demand flooding-based routing algorithms with Bluetooth are provided. Multi-hop networks with multiple piconet topologies in Bluetooth (scatternet) are used in the implementation. Results of the developed ad-hoc routing protocol are contrasted with the previous theoretical studies.

Chapter 6 focuses on Zigbee as an alternative technology to Bluetooth when deploying mobile WSNs. A theoretical overview of handover, routing and positioning support with Zigbee is reported. The chapter provides a quantitative and qualitative comparison against Bluetooth. Conclusions and ideas for future research are presented in Chapter 7.
Chapter 2

Introduction

The interest in sensor networks for military surveillance systems is the main reason of the performed work in wireless ad-hoc sensor networks. This work has been focused in the communication and computation trade-offs including their use in ubiquitous environments [11]. Several research projects have been conducted around the world in this area. One of the most important projects is the SensIT project [12] with focus on development of cheap, smart devices with sensing capabilities networked through wireless links. Civil applications for homes, cities, environment and home-care are covered under the scope of the SensIT project. In particular industrial and medical applications with focus on mobility ground the work in this thesis. Demands and added functionality for such applications have been investigated and two wireless technologies (Bluetooth and Zigbee) have been used for test and evaluation of mobile WSNs.

In this chapter WSNs are introduced and a brief discussion of mobile communications issues is made to highlight the design challenges of mobile WSNs. The following sections of the chapter provide an introduction to Bluetooth and Zigbee. Moreover an architectural description of the developed sensors in this work is also provided.

2.1 Wireless Sensor Networks (WSNs)

The huge growth of wireless communications in recent years is mostly due to new connectivity demands and advances in technology development of low power Complementary Metal-Oxide-Semiconductor (CMOS) transceivers. An example of the new demands is the increasing exchange of data in Internet services which has led to the deployment of wireless networks for data transmissions [6] characterized by the need of high data throughput. A typical example of these networks is represented by Wireless Local Area Networks (WLANs) IEEE Stds 802.11/a/b [3]. Wireless
Personal Area Networks (WPANs) usually supporting links up to 10 m in length, are another emblematic example. One of the best known WPANs is Bluetooth which is based on IEEE Std 802.15.1 [13]. On the other hand there are wireless network applications requiring low data throughput such as home automation and health monitoring [14]. Since most of these low-data-rate applications involve some form of sensing and actuation, networks supporting them have been designated as wireless sensor networks [6] due to the length of the name "Wireless Sensor & Actuator Networks" [4]. It is important to distinguish WSNs from Mobile Ad-hoc Networks (MANETs). MANETs share some characteristics with WSNs such as ad-hoc networking and low power consumption but they are different in the sense that they pursue different research goals [7].

2.1.1 WSNs Applications

An overview of the most relevant applications for WSNs is given in the following sections:

- **Industrial Control and Monitoring**
  Significant cost savings may be achieved with the use of inexpensive wireless sensors/actuators since sensors and actuators in industrial plants are often relatively inexpensive when compared with the cost of installed cable to communicate them. Examples include industrial safety, monitoring and control of rotating and/or moving machinery and heating, ventilating and air conditioning (HVAC) of buildings.

- **Home Automation and Consumer Electronics**
  Most of industrial applications have parallels in the home [15], for example a home HVAC system. Other applications include security systems, commercial lighting, PC peripherals and computer enhanced toys. Another interesting application is the use of location-aware capabilities of WSNs for consumer related activities such as tourism and shopping [16].

- **Security and Military Sensing**
  Thanks to WSN components specific characteristics such as, small size, unobtrusive and distributed mesh topologies; it is possible to deploy camouflaged sensors to resemble for example native rock or other nature bodies. The intrinsic resilient nature of WSNs makes them difficult to destroy in battle [17].

- **Asset Tracking and Supply Chain Management**
  Warehouses require efficient organization to be able to manage the stored items. With the help of WSNs item locations can be accurately identified. Tracking is also a field that could be beneficed by the use of WSNs. One tracking example is the Authenticated Tracking and Monitoring System (ATMS) [18].
2.1. WIRELESS SENSOR NETWORKS (WSNS)

- **Intelligent Agriculture and Environmental Sensing**
  WSNs may aid farmers gathering information about soil moisture, temperature, received sunshine and other related parameters. Vineyards are one of the first targeted markets for these kind of applications. Environmental sensing may be achieved with ultra low-power WSNs providing information about atmosphere contaminants, noise pollution and so on.

- **Health Monitoring**
  Health monitoring must be understood as monitoring of non-life-critical health information, to differentiate it from medical telemetry [6]. There are two principal applications in health monitoring, athletic performance monitoring and home health monitoring.

2.1.2 WSNs Design Challenges

To be able to accomplish the applications just mentioned a combination of technical challenges not found in other wireless networks including new communication approaches, new architectures and protocol concepts are required [7]. The characterizing technical challenges in WSNs are presented in the following sections.

- **Low Cost**
  Inexpensive products should not become expensive by adding wireless connectivity. Particularly in the case of applications with large number of devices such as WSNs. Ad-hoc networking, self-configuration and self-maintenance become then crucial challenges to reduce the large costs of network administration and maintenance.

- **Low Power Consumption**
  The more and more common use of portable devices with complete untethered Radio Frequency (RF) transceivers (no access to external power) requires the use of batteries or power scavenging [19]. Battery life requirements depend on the application. Having in mind that battery replacement goes against ease of installation and low-cost operation, a general assumption for certain industrial and medical sensors is that cell batteries powering the sensors should last from several months to many years. For example, an AAA battery with capacity of 750 mAh powering a RF transceiver consuming 10 mA (typical off-the-shelf active current consumption) will last for two years approximately if a duty-cycle of less than 0.5 % is maintained.

- **Range**
  RF power outputs are constrained by governmental regulations and implementation economics. Typical power outputs in unlicensed bands range from 0 dBm to 20 dBm. Limitations in power establish limited communication range. Multi-hop network routing protocols are then needed. An example of
a typical range for a RF output power of 0 dBm with radios having a sen-
sitivity of -70 dBm in the 2.4 GHz band gives a range of 10 m for average
indoor environments using the log-distance path-loss model given by (2.1)
with path-loss coefficient \( n \) equal to 3. \( PL \) in (2.1) stands for path-loss, \( d \)
refers to distance and \( d_0 \) stands for reference distance.

\[
PL(dB) = PL(d_0) + 10n \log_{10} \left( \frac{d}{d_0} \right).
\]  

(2.1)

• Worldwide Availability
Worldwide operation is achieved by using standards in the same frequency
band. This band has to be worldwide available thereby maximizing the total
available market for WSNs. The most common bands used (or planned to be
used) for WSNs are:

- 868.0 MHz - 868.6 MHz: Available in most European countries.
- 902 MHz - 928 MHz: Available in North America.
- 2.42 GHz - 2.48 GHz: Available in most countries worldwide.
- 5.7 GHz - 5.89 GHz: Available in most countries worldwide.

• Network Topology
WSNs usually offer the possibility of deploy star networks employing a sin-
gle master sensor and one or several slaves as depicted in Fig. 2.1 a). The
maximum number of hops in such topology is two. However due to output
power limitations this topology may not be optimal if larger areas are to be
covered. The solution to enlarge the physical area of the network is the use
of multi-hop routing techniques. Multi-hop networks require new topologies
like mesh or cluster types as depicted in Fig. 2.1 b).
2.1. WIRELESS SENSOR NETWORKS (WSNs)

- Security
  The main security goal in WSNs is not message encryption. Instead, ensuring that any message received has not been modified (integrity) and is from the sender it is supposed to be (authentication) are often the most important security goals. Security targets then to hinder eavesdroppers to inject fails or modified messages into the WSN.

- Data Throughput
  Data throughput is a measurement of the communications efficiency of the network and it is negatively affected by protocol overhead. Since control and status information commonly exchanged in WSNs is small compared to protocol overhead, the data throughput is significantly reduced compared with the raw data rate. Raw data rate is the data rate transmitted over the air, which may be significantly higher than data throughput. For example, Bluetooth signals data at 1 Mbps over the air but the maximum achievable data throughput is 723.2 Kbps [5].

- Message Latency
  WSNs do not usually demand deterministic behavior thus providing relaxed QoS, latencies of seconds or minutes are quite acceptable in many applications.

- Mobility
  Mobility is not a main requirement in WSNs in general. This fact implies that the WSN is released from identifying open communication routes, which is reflected in less control traffic overhead. However this thesis covers applications of mobile WSNs where ad-hoc routing techniques are required as it happens in MANETs.

- Localization
  The collection of techniques and mechanisms that measure spatial relationships are referred as localization [20]. Localization is an important factor in WSNs since sensors are coupled to the physical world, and they have spatial relationships to other objects and WSN members. One example is the collection of temperature readings in a warehouse without location information. The collected data is then only useful to compute the average temperature of the warehouse [21].

2.1.3 WSNs Projects, Standards and Platforms

WSNs have been a research topic since 1978, since then many projects have been conducted in the area. As previously mentioned, the SensIT project [12] started in 1998 is one of the most relevant projects. This project focuses on wireless ad-hoc
networks for large distributed military sensor systems. SensIT involved 29 sub-
projects from 25 different institutions. Some of the most relevant WSN works are
described below (not all funded though by SensIT).

- **Wireless Integrated Network Sensors (WINS)**
  Developed at University of California at Los Angeles. Results of this project
  have been recently commercialized by the Sensoria Corporation (San Diego,
  California) [22]. The conducted research ranges from Micro Electro Mechani-
cal System (MEMS) sensor and transceiver integration [23] to studies of prin-
ciples of sensing and detection theory. The data link layer is based on Time
Division Multiple Access (TDMA), the Physical Layer (PHY) uses spread
spectrum techniques [24], this technique is explained in section 2.4.2.

- **PicoRadio**
  The Picoradio project started in 1999 at Berkeley by Jan M. Rabaey with fo-
cus on ”the assembly of an ad-hoc wireless network of self-contained mesoscale,
low-cost, low-energy sensor and monitoring nodes [25]”. The PHY layer
uses Direct Sequence Spread Spectrum (DSSS) as well; the Medium Access
(MAC) protocol combines spread spectrum and Carrier Sense Multiple Ac-
cess (CSMA). To achieve this a sensor node randomly selects a channel and
checks it for activity. If the channel is occupied, the sensor node will select
another until an idle channel is found.

- **Smart Dust**
  The Smart Dust program aims to use MEMS-based motes small enough to
remain suspended in air by air currents with sensing and communication
capabilities lasting from hours to several days [26]. The physical layer is
based on passive optical transmission [27].

- **µAMPS**
  The µAMPS program at Massachusetts Institute of Technology (MIT) is fo-
cused on low power WSNs. The best known result in this program is the
Low Energy Adaptive Clustering Hierarchy (LEACH) protocol [28] which
randomizes the assignment of the cluster head role among multiple nodes in
order to reduce the high power consumption on a specific cluster and thereby
prolonging network lifetime.

- **Terminodes and Mobile Ad-hoc Networks (MANET)**
  The Terminodes project [29] and the MANETs Working Group of the Internet
Engineering Task Force (IETF) [30] are focused in the study of mobile ad-
hoc networks. The main focus of these works relies on the routing problem
generated when ad-hoc nodes are mobile.

- **The IEEE Std 802.15.4 Low-Rate WPAN Standard and Zigbee**
  The main purpose of the IEEE Std 802.15.4 is to provide ultra low complex-
ity, ultra low cost, ultra low power consumption and low data rate wireless
connectivity among inexpensive devices, with maximum and minimum raw data rates of 250 Kbps and 20 Kbps respectively [31]. A more detailed description of this standard and its implementation in Zigbee are presented in the Zigbee Basics section of this chapter. Note that IEEE Std 802.15.4 does not standardize the higher communication protocol layers, including the Network (NWK) and Application (APP) layers. The standard for these layers is provided by the Zigbee alliance [32].

- The IEEE Std 1451.5 Wireless Smart Transducer Interface Standard
  The IEEE 1451 family of standards provides a solution to produce transducers compliant with a large number of network communication protocols. The system involves standard Transducer Electronic Datasheet (TEDS) and Network Capable Application Processor (NCAP). TEDS is used to provide descriptions of transducers to measurement systems, control systems, and in general any device on the network. NCAP is the protocol handling processor between the sensor and the network. The latest IEEE Std 1451 (1451.5) is focused on wireless sensors [33].

- Platforms
  There are available platforms including all the required hardware, software and development tools for the deployment of WSNs. Some of the platforms are based on research projects while others are commercial. The following list presents some of the most relevant platforms including the developed platform in this thesis.
  
  - Motes Nodes
    Motes nodes are a family of embedded sensors developed at Berkeley. The architecture has a two-CPU design. The PHY layer on MICA Motes is based on the TR1000 chip-set from RF Monolithics Inc. operating at 916 MHz band with a maximum of 50 Kbps raw data rate. With transmission power control enabled, Motes give a maximum transmission range of 91.4 m.
  
  - Energy Efficient Sensor Networks (EYES) Nodes
    EYES nodes have been developed by Infineon in the context of an European Union sponsored project. EYES nodes have a single-CPU design, the main micro-controller (MCU), a Texas Instruments (TI) MSP 430 is used for general processing, the PHY layer is implemented with Infineon’s TDA-5250 radio modem. EYES nodes support transmission power control, measurements of signal strength, USB interface and the possibility to add additional sensor/actuators.
  
  - BTnodes
    The "BTnodes" [34] developed at the ETH in Zürich also feature a single-CPU design with an Atmel ATmega 128L MCU and 128 Kilo Bytes (KB)
FLASH memory. Unlike most other sensor nodes, the PHY layer is based on Bluetooth and a TI-Chipcon CC1000 radio operating between 433 MHz and 915 MHz.

- **Scatterweb**
  The Scatterweb platform [35] has been developed at the Freie Universität in Berlin. An entire family of nodes based on TI MSP 430 MCU has been developed. Applications range from embedded web servers to standard sensor nodes. Sensor nodes support a wide range of interconnection possibilities including Bluetooth, low power radios, Inter Integrated Circuit (I²C) and Control Area Network (CAN).

- **MDU Mobile Nodes**
  Developed at Mälardalen University as result of the implementation of biomedical wireless sensors with special focus on mobility support. The sensors are based on Bluetooth and Zigbee operating at 2.4 GHz band.

The latest architectures offer standard Bluetooth functionality (serial port profile) based on one-CPU and stand-alone designs. [36], [37], [38], [39]. For one-CPU Bluetooth solutions a Microchip MCU is used to implement the acquisition and the data processing and the physical layer is based on CSR BC2 modules. The Bluetooth stack has been deployed in two ways. The first one implements part of the upper Bluetooth layers in the MCU as illustrated in Fig. 2.2 a) and for the second one the stack implementing the Serial Port Profile (SPP) is embedded in the Bluetooth module as Fig. 2.2 b) illustrates. The latest Bluetooth design offers a stand-alone solution where the Bluetooth stack and the application are embedded in the Bluetooth module. This solution enables sensors with...
For Zigbee sensors, one-CPU solutions have been implemented. An Atmel ATmega 128L is used as MCU and the physical layer is implemented by TI-Chipcon CC2420 as Fig. 2.3 depicts.

The developed algorithms and protocols support local positioning, handover and ad-hoc routing.

– Commercial Nodes

There are a few certified commercial solutions for WSNs apart from the academic research prototypes. Some of the companies behind these commercial solutions are Ember, Millenial and Motorola with its recent neurRFon based on Zigbee.

It is important to note that the stand-alone Bluetooth solution developed in this work reduces size, cost and power consumption compared to other Bluetooth-based platforms. However there is a trade-off between the sensor size and offered functionality. The stand-alone design provides a sampling rate of 10 Hz with modifications of the SPP at RFCOMM level to support sensor data transmissions and handovers.

### 2.2 Mobile Wireless Communications

The definitions of mobile and wireless vary from person to person. Even though the terms mobile and wireless are two different things, sometimes they are used interchangeably [40]. The term wireless will refer in this thesis to the transmission of information over the air using RF techniques. Mobility on the other hand can be defined by two different terms, "user mobility" and "device portability". User mobility refers to a user that can access the same telecommunication services from
different locations, i.e., services will follow the user while moving. Call forwarding is a simple example of user mobility. On the other hand device portability refers to a communicating device that moves with or without a user. Device portability implies several mechanisms in the device and the network to ensure that communication is still possible while the device is moving. A typical example of device portability is the mobile phone system, where the system hands the device from one radio transmitter (also known as base-station) to the next if signal strength becomes too low [9]. Although WSN nodes are assumed to be substantially stationary with pervasive computing [41], mobility may still be a related field to WSNs if they are considered as mobile ad-hoc networks [42]. WSNs with mobility are intended for applications where the sensor node might not be constrained to a geographical area. These applications are characterized by device portability and ubiquitous [41] computing. The main issues addressed in mobile WSNs, apart from the conventional WSNs issues, are then handover and the routing problem created in MANETs.

2.2.1 Mobile Wireless Communication Applications

Despite of the technical challenges there is a wide range of mobile wireless applications, some of the most relevant applications follow.

- **Vehicles**
  Even though today’s cars already implement mobile wireless communications such as commercial radio, the trend is to implement more wireless communication systems and mobility aware applications that will provide music, information about road conditions, weather reports and other broadcast information to the driver [9].

- **Business**
  The concept of mobile office in a laptop is being adopted by many companies. The travelling worker uses his/her laptop from anywhere in the world but still can access the company’s database to ensure that fields on his/her laptop reflect the current situation.

- **Replacement of Wired Networks**
  In some applications such as weather forecasts and earthquake detection it is often impossible to wire remote sensors. Moreover trade show centers demand a highly dynamic infrastructure and historic buildings may not allow destroying valuable walls or floors to deploy wired networks. In these circumstances and also due to economic reasons wireless networks are more appropriate.

- **Location Dependent Services**
  Several applications need location awareness services for further activities.
Tourism and shopping applications may benefit from these services. For example a typical follow on application is the museum visitor where nomadic data streams follow the visitor depending on his/her location.

- **Mobile Wireless Devices**
  Many of electronic consumer products such as mobile phones, pagers and laptops support wireless mobility [9]. However in this thesis the main focus relies on mobile wireless sensor devices. One example could be a patient moving inside hospital facilities. The biomedical sensors attached to the patient body, e.g. Electro-Cardio-Gram (ECG), temperature and pulse are moving and performing handovers between hospital’s APs.

### 2.2.2 Mobile Wireless Communication Design Challenges

Mobile wireless communications differ from communications in traditional wired, fixed networks in many aspects such as medium access and interferences among others. These differences open up the following technical challenges:

- **Interference & System Capacity**
  Transmitted data losses and bit error rates (BERs) are usually higher in wireless communications since the medium is more likely to suffer interferences than wired networks where shielding adds robustness.

- **Frequency Reuse**
  The frequency spectrum is a limited resource, therefore an intelligent use of frequencies is needed. A typical example of frequency reuse is the cellular radio system where intelligent allocation and reuse of channels throughout a coverage region is used [43].

- **Handoff Strategies**
  Following with the above mentioned cellular example, when a mobile node moves into a different cell while transmission of data is in progress, there must be a transfer of the data to a new channel belonging to the new Base Station (BS) as Fig. 2.4 depicts. Handoff is then composed by two actions: identifying a new BS and allocation of a new channel with the new BS. Handoff also denoted as handover has been investigated and both hard and soft handoffs have been tested as described in Chapter 4.

- **Compatibility**
  In many cases mobility solutions have to be integrated into existing systems or at least work with them, e.g., mobile wireless Internet. The technical changes introduced by mobility solutions can not change the applications or network protocols already in use if compatibility must be accomplished.

- **Transparency**
  Mobility issues must remain "invisible" for many higher layer protocols and
applications. Even some often unavoidable effects like lower bandwidth and some interruption in service should not affect the higher level protocols in a way that communications are permanently stopped.

- **Scalability and Efficiency**
  Similar to the discussions for WSNs, introducing mobility to the system must not jeopardize the system efficiency. Keeping in mind that wireless links often have lower bandwidth than wired links too many new messages flooding the network should not be generated [9].

- **Security**
  As discussed in section 2.1.2 mobile systems also require at least authentication of all messages related to the management of mobile nodes.

- **Modulation Technique**
  Choosing the right modulation technique becomes crucial in mobile wireless communication systems. Mobile radios are usually based on Frequency Modulations (FM) since power consumption and robustness are better in this type of modulations than results obtained with Amplitude Modulation (AM) based techniques [44].

- **Mobile Radio Propagation**
  As Rappaport describes in ref. [44], several issues must be taken into account when deploying mobile wireless communications including reflections, diffractions, scattering, doppler spread and fading among others. These issues affect the BER of any modulation technique.
2.2. MOBILE WIRELESS COMMUNICATIONS

- **Reliability**
  The perceived and real reliability of wireless mobile applications can be lower than in wired networks. Wireless communications may not be enough to replace hardwired connections if high reliability is a design requirement [4].

- **Shared Medium Access**
  RF communications are performed via a shared medium. Therefore to gain access, different competitors have to "fight" for the medium. The challenge is then how to combine access, coding, and multiplexing schemes to provide QoS efficiently [45].

- **Ad-hoc Networking**
  Spontaneous networking is allowed in wireless and mobile computing without prior set-up of an infrastructure. The challenges are then how to implement routing on the NWK and APP layers, service discovery, network scalability, reliability and stability as described in ref. [9].

2.2.3 Mobile WSNs

One of the main virtues of wireless communication is its ability to support mobile participants, although in WSNs this capability is traded with "ease of installation". Still certain mobility concepts can be used to enable ad-hoc networking. In WSNs mobility can appear in three main forms according to ref. [7].

- **Node Mobility**
  Wireless sensors nodes are mobile in this context, the meaning of such mobility is highly application dependent. Node mobility implies that the network has to reorganize itself frequently, i.e., the logical topology of the network will change if just one of its members change its logical link due to a location change. There is then an obvious trade-off between the mobility level of sensor nodes and the energy required to maintain a desired level of functionality.

- **Sink Mobility**
  Sink mobility refers to mobile information sinks, which can be considered as a special case of node mobility. The most interesting aspect is the mobility of information sinks which are not part of the network, i.e., a mobile user which collects data via a Personal Digital Assistant (PDA) from a sensor network. The challenge is then the design choice for the appropriate protocol layers to support mobile sinks requesting data at a starting location and completing its interaction at a different location requiring the use of different network resources [46].

- **Event Mobility**
  This is a quite uncommon form of mobility. Event mobility refers to applications where event detection is required, particularly in tracking applications,
the cause of the events of the objects to be tracked can be mobile. The challenge is the design of network interactions to ensure that the observed event is covered by a sufficient number of sensors at all time. Moreover an area of activity within the network must accompany the mobile source event, this is also known as the frisbee model, introduced in ref. [47].

Handover, routing and location are the three major aspects included in mobility management [48].

**Handover**

Handover is the British English term for handoff. In cellular systems (Global System for Mobile Communications, (GSM) and Universal Mobile Telecommunications System (UMTS)), the term handover refers to the process of transferring an ongoing call or data session from one channel connected to the core network to another, these channels are usually from different BSs. Hard handover described as "break before make" (referring to the radio link) is the most basic form of handover [9]. Hard handovers are common in GSM and TDM or FDM related techniques. The main characteristic of hard handovers is that the mobile phone can only be connected to one BS at a time as Fig. 2.5 a) illustrates. The second type of handover is that used in Code Division Multiple Access (CDMA) systems also explained in
ref. [9]. The soft handover is described as "make before break". In this case the mobile phone supports two different radio links simultaneously as Fig. 2.5 b) shows. Handover must be implemented in mobile WSNs as a first step to provide mobility, in this thesis both hard and soft handovers are investigated and tested as described in Chapter 4 and Chapter 6.

There is another type of handover known as inter-technology handover (vertical handover) where a connection is transferred from one technology to another, e.g. a call being transferred from DECT to WLAN. This last type of handover is out of the scope of the work in this thesis.

Routing over MANETs

After (or while) a handover has been performed the next challenge is to route the information from/to the new BS. The study of routing over MANETs have provided several protocols such as Ad-hoc On Demand Distance Vector (AODV) and Dynamic Source Routing (DSR) proposed to provide robustness in the face of changing topologies [49]. Several mobile routers and associated hosts connected by wireless links forming an ad-hoc topology define a MANET. Routers must be able to move and self-configure themselves. The resulting network topologies are then dynamic and routing protocols must support this dynamic environment.

2.3 Bluetooth Basics

In 1994 Ericsson Mobile Communications AB started to investigate alternatives to connect mobile phones with external accessories. The result of these investigations led to the first Bluetooth specification designated as Bluetooth 1.0. Bluetooth uses radio links for exchange of data and speech between mobile phones, headsets and computing devices. The technology is named after Harald Blatand (Blatand is danish for Bluetooth), Harald was a Danish Viking king who unified Denmark and Norway. Since Bluetooth was expected to unify the telecommunications and computing industries, the name seemed to fit in [50]. Moreover the Bluetooth Special Interest Group (SIG) formed by companies cooperating to promote and define the Bluetooth specification was formed. Improvements have been added to newer versions of the specification like the latest Bluetooth v2.0 supporting Enhanced Data Rate (EDR) and Adaptive Frequency Hop Spread Spectrum (A-FHSS). Table 2.1 gives a general overview of the different Bluetooth versions including some characterizing parameters. In order to keep this text consistent it should only be considered "Bluetooth v1.1” as the version of the implemented work, thereby Bluetooth will be used as a synonym of "Bluetooth v1.1” if not otherwise specified.
Table 2.1: Bluetooth specifications [51].

<table>
<thead>
<tr>
<th>SS Technique</th>
<th>v1.0/b</th>
<th>v1.1</th>
<th>v1.2</th>
<th>v2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulation</td>
<td>FHSS</td>
<td>FHSS</td>
<td>A−FHSS</td>
<td>A−FHSS</td>
</tr>
<tr>
<td>GFSK</td>
<td>GFSK</td>
<td>GFSK</td>
<td>GFSK</td>
<td>GFSK</td>
</tr>
<tr>
<td>π/4 DQPSK</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>8DPSK</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>Raw Rate</td>
<td>1 Mbps</td>
<td>1 Mbps</td>
<td>1 Mbps</td>
<td>1 Mbps</td>
</tr>
<tr>
<td>RSSI</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HCI</td>
<td>−</td>
<td>−</td>
<td>3W−UART</td>
<td>3W−UART</td>
</tr>
<tr>
<td>Expires</td>
<td>17−1−2007</td>
<td>9−8−2007</td>
<td>−</td>
<td>−</td>
</tr>
</tbody>
</table>

2.3.1 Bluetooth Overview and Features

Bluetooth is a RF wireless technology with many components and abstraction layers. Bluetooth is originally intended to replace cable(s) connecting portable and/or fixed electronic devices such as mobile phone handsets, headsets, and portable computers. The basic Bluetooth features are low power, low cost and short-range. Short range refers to a person’s operating space or Personal Area Network (PAN) (typically 10 m). The main purpose is that devices will communicate seamlessly supporting Asynchronous Connection Less (ACL) and Synchronous Connection Oriented (SCO) links for data and voice respectively. The Bluetooth specification is open and global, detailing the complete system from the PHY layer up to the APP layer. Bluetooth operates in the globally available, license-free Industrial Scientific and Medical (ISM) band at 2.4 GHz. Mature versions of the Bluetooth specification used a Gaussian Frequency Shift Keying (GFSK) modulation scheme signalling data at 1 Mega-symbol per second obtaining the maximum available channel bandwidth (1 MHz) as exposed in Table 2.1. However the maximum achievable throughput with specification 1.0b, 1.1 and 1.2 is 723.2 Kbps as (2.2) describes [5].

\[
\text{Throughput} = \frac{\text{Max. User Payload}}{6 \cdot t_{\text{slot}}} = \frac{339 \text{Bytes} \cdot 8 \text{bits}}{6 \cdot 625 \mu\text{s}} = 723.2 \text{ Kbps.} \quad (2.2)
\]

The newest version (v2.0 + EDR) on the other hand, increases raw data-rates up to 2 Mbps and 3 Mbps allowing users to run several links concurrently by enabling sufficient bandwidth. The increase in raw data rate has other advantages apart from the just mentioned, transceivers need only to remain fully active for about a third of the time (for transmitting the same amount of data) and thereby battery life
is prolonged. This increment is accomplished adding two modulation techniques: \( \pi/4 \) Differential Quadrature Phase Shift Keying (\( \pi/4 \) DQPSK) and Eight Phase Differential Phase Shift Keying (8DPSK) which are described in detail at the specification core document. Note that Bluetooth v2.0 still uses GFSK for transmitting the access code and header of packets in order to keep backwards compatibility with previous versions. Bluetooth uses Time Division Duplexing (TDD) as a medium access protocol with a FHSS technique [50]. This implies that Bluetooth front-ends have to re-synchronize to a new frequency channel after a new hop. The hopping sequence is pseudo-randomly generated resulting in a robust technology against interference and noise. The duration while the frequency is stable is called a slot. Each slot is 625 \( \mu s \) long as Fig. 2.6 illustrates. Bluetooth is designed to be a low power technology allowing portable applications, the radio power must be limited. Therefore three different power classes are supported providing approximately 10 m, 20 m and 100 m range respectively [5]. The three power classes in Bluetooth correspond to: class 1 device (0 dBm to 20 dBm) range 100 m, class 2 device (-6 dBm to 4 dBm) range 10 m, and class 3 device (up to 0 dBm) range 1 m.

### 2.3.2 Bluetooth Protocol Stack

Bluetooth defines a protocol stack to enable a wide variety of applications to interact. These applications may reside in devices from different manufacturers. The Bluetooth specification ensures that such different devices can find each other, discover the offered services and use them. The Bluetooth protocol stack does not match the Open Systems Interconnect (OSI) standard reference model. The

---

**Figure 2.6:** Bluetooth slot timing.
protocol stack is defined as a series of layers as depicted in Fig. 2.7, though there are some features which cover several layers. Middle layers hiding specific Bluetooth functionality are provided by the specification in order to support familiar data formats, and protocols, and integrate Bluetooth into existing applications, e.g., in Fig. 2.7 internet applications supporting TCP/IP protocols will be adapted to Bluetooth baseband with Radio Frequency Communications (RFCOMM) and L2CAP layers [5]. The essential layers to Bluetooth are at the bottom of the stack including Physical (PHY), Baseband (BB), Link Manager (LM), Logical Link Control, L2CAP and Service Discovery Protocol (SDP). Above these layers, different applications require different selections from the higher layers [5]. Moreover the Bluetooth specification defines a set of profiles. A Bluetooth profile is a set of actions providing how applications should use the Bluetooth protocol stack. The main Bluetooth protocol layers are:

- **Physical Layer (PHY)**
  The Bluetooth PHY layer defines the requirements for a Bluetooth transceiver operating in the 2.4 GHz ISM band modulating and demodulating data for transmission and reception on air [51].

- **Baseband (BB)**
  The BB layer is in charge of bitstream processing immediately before and after RF transmission. Forward Error Correction (FEC) and Automatic Repeat Request (ARQ) are some of the main functions performed at this layer.
• Link Manager (LM)
  LM layer is used for link set-up and control.

• Host Controller Interface (HCI)
  HCI is intended to be used between a host and a Bluetooth device. HCI
  provides a command interface to the BB, LC and LM, and access to hardware
  status. Several HCI transport interfaces are supported, such as USB, RS-232
  or Universal Asynchronous Receiver Transmitter (UART).

• Logical Link & Adaptation Protocol (L2CAP)
  The L2CAP layer multiplexes higher level protocols and data from higher
  layers and performs packet segmentation as well as reassembly. The L2CAP
  protocol is also in charge of QoS requirements like packet size, latency and
  maximal data rate.

• Radio Frequency COMMunications (RFCOMM)
  RFCOMM is a protocol based on the European Telecommunications Stan-
  dards Institute (ETSI) standard TS 07.10, it provides emulation of serial
  ports over the L2CAP layer. It supports multiple concurrent connections
  over the same radio link using a multiplexing technique.

• Service Discovery Protocol (SDP)
  SDP is used to aid applications in discovering Bluetooth features in remote
  devices. Examples of services are serial port support and object exchange
  among others.

• Profiles & Applications
  Different profiles exist supporting a set of applications, in this thesis the serial
  port profile (SPP) and the PAN profiles are the most interesting due to their
  applications for WSNs.

2.3.3 Bluetooth Network Topologies

Bluetooth supports a simple star topology known as piconet with one master device
and seven active slaves. Another supported topology is referred to as scatternet
where several piconets are interconnected. Before describing these two topologies
lets study the two mechanisms involved in the creation of a Bluetooth link (a basic
piconet).

Discovering Devices: Inquiry

Inquiry and inquiry-scan modes are required before two devices can discover each
other. The inquiring device tries to discover devices in its neighborhood, and the
inquiry-scanning device is willing to be discovered. The inquiring device transmits
inquiry packets. These short packets are sent in a sequence of different frequencies. The inquiring device changes its frequency 3200 times per second with the goal of cover a range of frequencies as rapidly as possible. An inquiry access code is sent in the short packets which inquiry-scanning devices will recognize. On the other hand the inquiry-scanning device changes frequencies very slowly: once every 1.28 s. Due to the different speeds changing frequency, the two devices will ultimately meet on the same frequency. At that point scanning-devices will respond to inquiries by sending a Frequency Hop Synchronization (FHS) packet.

Connecting Devices: Paging

Two devices must be in page and page-scan modes respectively before creating a connection; the paging device (usually targeted as a master) starts the connection and the page-scanning device (normally targeted as a slave) responds. The paging device must know the identifier (ID) of the page-scanning device before it can create a connection. The ID from the page-scanning device corresponds to the 48-bit Bluetooth Device Address (BDA). The BDA, identifies a device uniquely; if the device is a master, the hopping sequence together with the link timing are obtained from its BDA.

Bluetooth Piconet

A piconet is the basic link in Bluetooth. A piconet is always formed by one master device and one or several slave devices. The network topology is a basic star with the master placed at the center of the star as depicted in Fig. 2.8. Slaves are addressed by their Active Member (AM) addresses composed by 3 bits. This implies that a maximum of seven slaves can remain active in a piconet, the 8th address is reserved for master broadcasting. Other slave devices supporting low power operation modes can remain as non-active members of the piconet in a so called parked state. The maximum allowed number of parked slaves is 255. Both ACL and SCO
2.3. BLUETOOTH BASICS

The Bluetooth clock is a 28-bit counter which is reset to 0 at power up. This real time clock synchronizes most Bluetooth operations, and it is incremented every half slot, or 312.5 µs. Every device has its own Native Clock (CLKN). A master controls the piconet and uses its CLKN as its internal reference timing to synchronize the slaves. Slaves must then synchronize their clocks to the master’s CLKN so transmission (TX) and reception (RX) slots are correctly lined up with the master’s. All this means that Bluetooth uses Time Division Multiplexing (TDM) to implement communications. It is important to remark that all data/voice transactions are always started by the master as shown in Fig. 2.9. Slaves are only allowed to transmit information to its master after a master’s transmission slot. Masters always transmit on even slots and slaves reply on odd-numbered slots [50]. Masters regularly poll their slaves to maintain QoS, if the master has nothing to send but it is still willing to receive data from a slave, the master transmits then a NULL packet as it happens in slots 0 and 2 in Fig. 2.9. If a SCO link is in operation, then that slave must be communicated regularly according to the SCO repetition rate. Polling scheduling in its basic form is accomplished in a round robin fashion.

**Bluetooth Scatternet**

The term scatternet refers to a network formed by two or more piconets interconnected. The main difference from a piconet is the existence of more than one master in the network as shown in Fig. 2.10. This in turn implies several channels with their own CLKNs and Frequency Hop Synchronization (FHS). Both masters and slaves may participate simultaneously in more than one piconet at a time, the main

![Figure 2.9: Bluetooth piconet operation with two slaves.](Image)
requirement they must fulfill is to re-synchronize to the FHS and CLKN of the targeted piconet. Switching in time bases will require guard-time to synchronize to the new time-base. In the worst case, there could be a whole slot pair out of sync as Fig. 2.11 illustrates. The Bluetooth specification does not define how scatternets should be formed neither how they must be managed. ACL links are supported in scatternets. SCO links in devices being members of more than one piconet should be carefully implemented since guard slots and re-synchronization may result in QoS reduction. The theoretical maximum number of nodes in a scatternet is as big as the user requires, however management and routing overheads impose a practical limit based on several trade-offs. As described in ref. [52] mobile sensors may be either configured as slaves or masters. The role election for a sensor is based on supported QoS of the network. When sensor nodes are configured as slaves, better channel sharing is achieved. For example, if two wireless sensors configured as slaves are connected to one AP configured as master, the resulting piconet is based on a single channel. On the other hand, if the AP is configured as a slave, then the AP involves itself into two piconets forming a scatternet, thus reducing its bandwidth capacity [53]. However if for some reason or design constraints the sensor must be initially configured as a master then a master-slave role exchange is required if a slave role is desired for the sensor.
2.3.4 Bluetooth Voice and Data Packets

Bluetooth packets can be single-slot data, three-slot data, and five-slot data. There are 1600 slots in a second, which means that each slot is 625 $\mu$s long. ACL packets are basically divided in Data-Medium rate (DM) and Data-High rate (DH). The main difference between DM and DH packets is that DM packets are more robust with lower throughput due to FEC overhead as shown in Table 2.2. When a Bluetooth link is established, the two involved transceivers negotiate the packet type they will use. Results of the negotiation allow symmetric and asymmetric data rates. Most of applications require asymmetric data rates. For example the maximum data throughput of 723.2 Kbps can only be achieved with 5 slot packets in one direction and single-slot packets in the opposite direction at 57.6 Kbps. In SCO links the slots may be spaced to use every slot pair, every second slot pair, or every third slot pair. Therefore the maximum number of SCO links supported in one piconet is three since SCO slots are reserved and can not be delayed.

2.3.5 Mobile WSN Support in Bluetooth

Is Bluetooth an appropriate technology to implement WSNs? The answer to this question varies from author to author.

On the one hand some WSNs experts and authors affirm that power consumption in Bluetooth is too high to be used in WSNs as Callaway affirms in ref. [6]. Moreover H. Karl affirms in ref. [7] that the need of a master which constantly polls its slaves is a notable drawback together with the fact that only seven active slaves are allowed in a piconet, a fact that limits the use of Bluetooth on dense WSNs.
Table 2.2: Bluetooth packet types [51].

<table>
<thead>
<tr>
<th>Type</th>
<th>Header (bytes)</th>
<th>Data (bytes)</th>
<th>FEC</th>
<th>CRC</th>
<th>Symmetric Max Rate (Kbps)</th>
<th>Asymm. Max Rate Forward (Kbps)</th>
<th>Asymm. Max Rate Reverse (Kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM1</td>
<td>1</td>
<td>0-17</td>
<td>2/3</td>
<td>yes</td>
<td>108.8</td>
<td>108.8</td>
<td>108.8</td>
</tr>
<tr>
<td>DH1</td>
<td>1</td>
<td>0-27</td>
<td>NO</td>
<td>yes</td>
<td>172.8</td>
<td>172.8</td>
<td>172.8</td>
</tr>
<tr>
<td>DM3</td>
<td>2</td>
<td>0-121</td>
<td>2/3</td>
<td>yes</td>
<td>258.1</td>
<td>387.2</td>
<td>54.4</td>
</tr>
<tr>
<td>DH3</td>
<td>2</td>
<td>0-183</td>
<td>NO</td>
<td>yes</td>
<td>390.4</td>
<td>586.6</td>
<td>86.4</td>
</tr>
<tr>
<td>DM5</td>
<td>2</td>
<td>0-224</td>
<td>2/3</td>
<td>yes</td>
<td>286.7</td>
<td>477.8</td>
<td>36.3</td>
</tr>
<tr>
<td>DH5</td>
<td>2</td>
<td>0-339</td>
<td>NO</td>
<td>yes</td>
<td>433.9</td>
<td>723.2</td>
<td>57.6</td>
</tr>
</tbody>
</table>

From the complexity point of view, Bluetooth devices are complex since they may take the role of masters and slaves, the fast frequency hopping technique also adds complexity in terms of tight synchronization.

On the other hand other authors have implemented WSNs with Bluetooth as the work performed in Zurich [34] describes. The last group of authors rely on two Bluetooth characteristics for the implementation of WSNs, the low power modes and the ad-hoc networking capabilities of Bluetooth respectively. These two characteristics are explained below.

**Bluetooth Low Power Modes**

One of the main challenges in WSNs based on Bluetooth is how to prolong sensor’s battery life since a Bluetooth radio can use up to 30 mA when receiving [50]. The main sources for power consumption are the front-end and the Voltage Controlled Oscillator (VCO). Therefore there is the possibility of switching off the high accuracy VCO and run by a less consuming VCO which also has less accuracy (only applicable in hold and park modes). The fact that sensors in infrastructure networks are usually configured as slaves increases their power consumption since sensor receivers are constantly on "listening" for packets that may be addressed to them [5]. To be able to save power in such sensors, the time that the receiver is on must be reduced, this is accomplished by means of using Bluetooth low power modes. Bluetooth supports three low power modes:

- **Park**
  
  In this mode a Bluetooth slave achieves the greatest power saving. Devices in this mode can not send and receive neither SCO or ACL information. The device loses its active member address so the master can connect to more
slaves, however the slave remains synchronized and wakes up periodically in beacon slots previously negotiated when parking took place.

- **Sniff**
  Devices in this mode support periodic SCO and ACL packets. The devices wake up periodically in sniffing slots but they keep their active member addresses so they can transmit to the master.

- **Hold**
  The slaves put their connections to a low power state for a single period. In this mode ACL packets are not supported.
  In these two last modes the slave and the master remain synchronized but they save power by switching off their radios by predefined periods [51].

It can be concluded that sniffing is the best low power mode for applications where the sensor regularly transmit (or receive) data. Another positive aspect in Bluetooth referred to power consumption is the use of power control. Transmit Power Control (TPC) is a method (mandatory for Class 1 devices) that allows dynamic configuration of the radio output power. When a device notices that the Received Signal Strength Indicator (RSSI) is outside the golden range limits, it issues an output power request to the remote device. The remote device then increments or decrements its output power one step following the predefined steps in the power table. This explains the fact that power consumption is dependent on the distance between two Bluetooth radios.

**Bluetooth Ad-hoc Networking**

Ad-hoc networking is described by the SIG working group responsible for ad-hoc networking in the scope of the PAN profile, this profile addresses Internet applications supporting the TCP/IP protocol stack. IP traffic is encapsulated in L2CAP packets payloads using the Bluetooth Network Encapsulation Protocol (BNEP). The group ad-hoc networking covers devices which connect to one or more PAN members, forwarding packets between PAN members when more than one is connected. The BNEP provides a way for IP packets to be routed to and from Bluetooth devices. The PAN profile allows slaves to communicate directly with one another (with multi-hop routing) [51].

**Bluetooth Mobility Support**

The Bluetooth LMP performs link supervision to detect events such as if the device at the other end has moved out of range. If this has happened a link supervision timer will elapse and allow the link to be shut down, so that its active member address can be reused. The logical solution for the remote end device is to involve itself in a handover mechanism if it is willing to still form part of the network. However handover is not supported by Bluetooth as J. Bray and C. F. Sturman
describe in ref. [50] and therefore the effective range of a network is constrained to the range of a piconet. Moreover the Bluetooth specification does not provide support for location and ad-hoc routing.

2.4 Zigbee Basics

Zigbee is a new wireless technology included in the WPANs scope. As Bluetooth, Zigbee is a short-range wireless networking technology where little or no infrastructure is required (no network setups and no APs) providing ubiquitous, untethered, short-range communications. Its main objective is to enable low cost, low power, reliable devices for monitoring and control purposes. The Zigbee specification provides a platform for implementation of wireless networked devices. Moreover the specification defines the network and stack models providing the framework to allow a separation of concerns for the specification, design and implementation of Zigbee devices.

2.4.1 Zigbee Overview & Features

The IEEE Std 802.15.4 is property of the Institute of Electrical and Electronics Engineers (IEEE) and is part of the IEEE Std 802.15 working-group. This working-group is in charge of the development of WPANs standards. The first result of the group was the IEEE Std 802.15.1 commonly known as Bluetooth, which is focused on cable replacement for consumer electronic devices. The second (IEEE Std 802.15.3) focuses on high-speed WPANs. Finally the third standard (IEEE Std 802.15.4) was defined as a standard to provide ultra-low complexity, low-cost, and extremely low-power wireless connectivity for inexpensive and portable devices [4]. The 802 standards only define the first two layers of the International Standard Organization (ISO)/ OSI protocol reference model. In other words, only the PHY layer and the data link layer are provided by the IEEE Std 802.15.4. The upper layers are not specified in the standard and are normally specified by a consortium formed by several manufacturers, distributors and users. In the case of IEEE Std 802.15.4 the consortium responsible for the development of the NWK and APP layers, and profiles is called Zigbee Alliance. The Zigbee Alliance is a trademark property of Phillips Corporation. Although the current trend in wireless networking is to provide higher data-rates and QoS, Zigbee is intended for low cost and low power short-range communications.

2.4.2 Zigbee Protocol Stack

Zigbee is a standard based on IEEE Std 802.15.4 which means that it uses the IEEE standard to build a bigger protocol stack as Fig. 2.12 illustrates. The protocols of the Zigbee stack are explained in the following sections:
2.4. ZIGBEE BASICS

Figure 2.12: Protocol stack of a Zigbee enabled node.

Table 2.3: EEE Std 80.15.4 frequency bands and modulation parameters [4].

<table>
<thead>
<tr>
<th>Band</th>
<th>Freq Bit Rate</th>
<th>Symbol Rate</th>
<th>Modulation</th>
<th>Chip Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>868 MHz</td>
<td>868-868.6</td>
<td>20</td>
<td>20</td>
<td>BPSK</td>
</tr>
<tr>
<td>915 MHz</td>
<td>902-928</td>
<td>40</td>
<td>40</td>
<td>BPSK</td>
</tr>
<tr>
<td>2400 MHz</td>
<td>2400-2835</td>
<td>250</td>
<td>62.5</td>
<td>O-QPSK</td>
</tr>
</tbody>
</table>

- Physical Layer (PHY)
  The PHY layer handles the signals, which are transmitted through the air. Modulation, DSSS and filtering are some of the operations performed by this layer. Zigbee is a spread spectrum technique as Bluetooth, which means that the information is spread in frequency when transmitted as opposed to narrow band communications where the information is constrained to a single channel. The IEEE Std 802.15.4 defines three different Industrial Scientific Medical license free bands as described in Table 2.3. It is important to remark that investigations and experiments in this thesis have been conducted with Zigbee modules in the 2.4 GHz band therefore Zigbee is a synonym of Zigbee 2.4 GHz in this thesis.

  - Modulation
    The modulation scheme must be efficient with a low-cost implementa-
CHAPTER 2. INTRODUCTION

Figure 2.13: DSSS modulator.

tion, therefore half duplex technique is used, which means that transmitter and receiver cannot be active simultaneously. The modulation scheme used in the 2.4 GHz band is Offset Quadrature Phase Shift Keying (O-QPSK).

- Direct Sequence Spread Spectrum (DSSS)
  DSSS is a wide-band technique, this means that spectrum utilization is higher but the signal to noise ratio is improved. The DSSS process is applied before data modulation in the transmitter and after modulation in the receiver; it consists of another phase modulation at a higher rate of the original data-rate. The 32-chip pseudo-random sequence to be transmitted is split between the orthogonal I and Q channels of the O-QPSK modulator, the overall chip rate is 2 Mchips/s as described in Table 2.3. However the chip rate in either I or Q channels is 1 Mchip/s. The DSSS sequence is achieved multiplying every single information bit by a chip sequence, the chip sequence is composed by 32 chips as illustrated in Fig. 2.13 where PN is the pseudo random sequence. The effect in frequency is that the base-band spectrum is expanded like depicted in Fig. 2.14 where Tb is the bit period, Tc is the chip period, Rb is the bit-rate and Rc is the chip-rate. In that way the spectral amplitude is reduced. This fact adds robustness against interferences since at the receiver a correlation is performed with a similar sequence to that used in the transmitter. Cost is reduced as well due to this ”process-gain” which reduces filtering requirements. An advantage of DSSS is that security is improved. Since spectral power density is lower it becomes harder to eavesdrop. Another advantage is that undesired signals are further spread to a wider bandwidth in the receiver and only the original data-modulated signal is dispread back [4].

- Transmitter Power
  The IEEE Std 802.15.4 requires that a compliant device must be able to transmit -3 dBm or 0.5 mW. This value is well-suited for instantaneous power capacity of inexpensive batteries and the capability of highly integrated low-cost System-on-a-Chip (SoC) implementations.
2.4. ZIGBEE BASICS

Figure 2.14: DSSS modulation.

- Receiver Sensitivity
  The standard defines a minimum sensitivity of -85 dBm in the 2.4 GHz Band. This value is well suited for low-cost receiver designs with little radio-frequency amplification. Radio-frequency amplification is a high power consuming stage in a receiver.

- Medium Access Layer (MAC)
  The MAC layer together with the Logical Link Control sub-layer form the ISO/OSI layer 2. The MAC layer is responsible for controlling the access to the shared medium and adds reliability to communications. This layer is the top layer defined by the IEEE Std 802.15.4. The MAC layer provides services to upper layers to create star topology networks, connections, acknowledgements (ACKs) (every transaction at MAC level is acknowledged) and security control.

- Accessing the Channel
  Before explaining access to the channel it is important to understand the concept of beacon. A beacon is a periodic packet transmitted by the PAN coordinator with information of the PAN-ID and synchronism. The use of beacons is a wise solution to save battery since devices can switch
off their radios between beacons. In Zigbee, the medium access method is Carrier Sensing Multiple Access with Collision Avoidance (CSMA-CA). CSMA-CA is a simple method that consists of listening before transmitting. CSMA-CA is used in every single contention slot. Contention slots are used within superframes and superframes are placed between consecutive beacons. The structure of a superframe is depicted in Fig. 2.15. Beacons are sent periodically, the information content in the beacon includes synchronization, beacon periodicity, PAN-ID and superframe structure. As depicted in Fig. 2.15 the contention access period is the time when CSMA-CA method is used. This is a very important issue since Bluetooth is a time division system, which guarantees access to the channel in a deterministic way. In this way Bluetooth is better suited for low latency applications. However it is possible to ensure a constant QoS in Zigbee with the use of Guaranteed Time Slots (GTSs). This means that after a beacon is sent, there is a contention period but at the end of the superframe there is place for GTS, these guaranteed slots are contention free so CSMA-CA method is not needed and they are granted by the PAN coordinator with previous request from a device. The length of the GTSs is also known as Contention Free Period (CFP). For non-beacon-enabled networks CSMA-CA is always used and the main drawback is that the coordinator (and routers) must be mains powered since its receiver must be constantly on.

Figure 2.15: Zigbee superframe structure a) without GTSs and b) with GTSs.
Network (NWK)
The network layer is responsible for the creation and maintenance of networks providing addressing and routing capabilities. Self healing is also implemented at this layer, in case of a node failure the network re-routes the message if alternative route paths are available. It is important to notice that networking capabilities are out of the scope of IEEE Std 802.15.4 and they are defined by the Zigbee Alliance. There are three Zigbee network routing types:

- Star Network Routing
  One coordinator with one or several end-devices (up to 65534).

- Tree Routing
  This is a "netmask" hierarchical style routing down or up the tree based on destination.

- Mesh Network Routing
  It is based on a modified version of Ad-hoc On Demand Distance Vector (AODV) originally developed by IETF in the scope of MANETs. This algorithm uses flooding to find out the routes from any source to any destination in the mesh. The routes are determined upon received route replies. The routes are stored in record tables.

Only Full Function Devices (FFDs) described in section 2.4.3 can issue a route discovery command in mesh networks.

Zigbee Application Layer
Each Zigbee device is defined by a set of endpoints (maximum 240). The endpoints are the different functionalities implemented at APP level. The endpoint 0 is reserved and corresponds to the Zigbee Device Object (ZDO). Each endpoint is described by a descriptor. A descriptor contains information about the clusters of the endpoint. A cluster is defined as a "shared variable" in the network and they are used to connect endpoints from different sensor nodes. In order to connect an endpoint outgoing cluster with another endpoint incoming cluster a binding process is required. Binding can be performed in two ways, on the one hand direct binding is used when the destination address is previously known, the PAN coordinator is then not required. On the other hand indirect binding is used when the PAN coordinator is needed to perform the operation, once binding is performed the PAN coordinator will always be required to accomplish data transmissions. Application message acknowledgement is also implemented at this layer and it is intended to add reliability in multi-hop communications. For example in Fig. 2.16 each device has two objects, one for the ZDO and one for the APP. The APP object has four clusters (two out, two in), the figure also shows a bind between an output cluster from the light switch node and an input cluster at the light controller node.
- Zigbee Profiles
  Zigbee application profiles ensure inter-operability. They define sets of rules to govern the communication sessions within devices manufactured by different providers. A profile provides information about the allowed device types, endpoints and clusters.

2.4.3 IEEE Std 802.15.4 Overview

The standard is intended for low power, low cost and low complexity wireless applications. Low power must be then be achieved at both receiver and transmitter sides. Low cost is achieved by several trade-offs in performance together with low administrative costs due to the use of ISM unlicensed bands [4].

Duty Cycle

Duty cycle in IEEE Std 802.15.4 may be as low as 2.16 ppm when working in beacon mode fashion. However, the standard supports non-beacon modes in a master-slave fashion where slaves may remain in stand-by mode until an external event occurs. They will then contact the master. However the master should be constantly listening which implies that the master should be main-powered [4].

Quality of Service

IEEE Std 802.15.4 does not provide isochronous communication, this means that communication is not periodically synchronized. However if low-latency communications are required, the PAN coordinator may grant GTS to avoid channel access delay. The PHY layer provides a maximum of 250 Kbit/s for the 2.4 GHz band, however the data throughput might be significantly reduced depending on the used duty-cycle. These two aspects: low data throughput and high latencies might be
undesired when implementing certain telemetry applications where constant latencies and relative high data-throughput are required.

Network Devices
An IEEE Std 802.15.4 network is composed by the following types of devices:

- Network Coordinator
  All IEEE Std 802.15.4 networks must have one PAN coordinator. The coordinator is the first device in the network; its primal functions are to establish a Personal Area Network Identifier (PAN-ID) and to choose the channel for communications.

- Reduced Function Device (RFD)
  Commonly referred as end-devices, they represent the minimum form of an IEEE Std 802.15.4 device with minimal functionality and thereby maximum battery life. End-devices cannot work as routers or bridges with other technologies.

- Full Functional Device (FFD)
  These devices are more complex in functionality demanding higher power consumption. A FFD can work as a router, bridge or PAN coordinator.

Connecting and Creating a Network
A Zigbee network is established executing the following steps: A FFD do an active channel scanning, in this way, the FFD scans the sixteen channels looking for free available channels. If a free channel is found then the FFD decides to become a PAN coordinator by establishing a PAN-ID and selecting the active channel. This feature is also known as Clear Channel Assessment (CCA) and it is used to avoid other wireless networks working in the same band of frequencies. The PAN coordinator starts sending periodic beacons announcing its network if a beacon-enabled network is required. If a non-beacon-enabled network is required, the coordinator will then listen constantly for incoming beacon requests from routers or end-devices. Once FFDs and/or RFDs devices have gathered the PAN information contained in a coordinator beacon, they can join the network by sending an association request to the coordinator. However the PAN coordinator decides in last instance if the new device will be associated (connected).

2.4.4 Zigbee Network Topologies
IEEE Std 802.15.4 offers different possibilities to create the following network topologies (understanding topology as the logical structure of a network) among others.
**Star Topology**

This is the basic topology and consists of a master which is the PAN coordinator, the remaining devices are slaves and connected as Fig. 2.17 depicts. The main advantages of this topology are the low latencies (2 hops maximum route path) and the ease of the routing algorithm. A drawback of this topology is the fact that all devices must be inside the PAN coordinator range.

**Tree Topology**

In this topology several devices may be connected to a single parent device but they can be connected to several children as Fig. 2.18 illustrates. In this topology the coordinator is only responsible for starting the network and choosing certain network parameters but the network may be extended through the use of routers. For example in Fig. 2.18 the device with address 1 is child of the PAN coordinator and parent of device 3, device 1 has also 3 grandchildren with addresses 4, 5 and 6 respectively. The routing algorithm must be more advanced than in a star topology network to support multi-hop routing but still the use of a tree structure simplifies the routing algorithm reducing it to a hierarchical routing strategy. For example in Fig. 2.18 all devices with addresses that are equal or higher than 3 are always routed via device 1 from the PAN coordinator.
2.4. ZIGBEE BASICS

Mesh Topology

This topology is possible when using mesh networks, in a mesh network a device can connect with any other device in its range (only applicable to FFDs). As the number of devices grows in a network it also increases the memory requirements to save addresses in the routing tables. The network can be also extended by the use of routers as Fig. 2.19 illustrates. The routing algorithm enables exchange of data between any pair of network devices allowing multiple routes. In this case the routing algorithm is more complex and can not be reduced to a hierarchical strategy as it happened with tree topologies.

Zigbee Device Types Model

Zigbee divides network devices in three categories:

- **Application Device Type**
  This category refers to the specific functionality of a sensor/actuator. E.g., a node can be a light or a remote switch.

- **Zigbee Logical Device**
  This category makes differences according to the network logical topology. For example, a device in a Zigbee network may adopt the coordinator, router or end-device roles.

- **Zigbee Physical Device Type**
  This category is associated to the physical properties of a device, the two supported types are FFDs and RFDs respectively as explained in section 2.4.3.
2.4.5 Mobile WSN Support in Zigbee

Zigbee presents better ad-hoc capabilities than Bluetooth in terms of self-formation and self-maintenance of large dense WSNs. The Zigbee network layer supports ad-hoc routing algorithms thus Zigbee is classified as an ad-hoc multi-hop WSN [32]. Zigbee still being closer to MANETs than Bluetooth it is not 100% a mobile ad-hoc WSN. The Zigbee routing protocol is a simplified version of AODV. This is an IETF MANET submission. The protocol uses a flooding algorithm to determine paths from source to destination in the mesh. Route replies determine viable paths in the mesh and routing tables are employed to record known paths. Zigbee is originally intended for industrial control, building automation and personal health care [4]. Thus Zigbee is a WSN technology supporting low power modes and ad-hoc networking. The remaining question is how well suited is Zigbee for mobile applications?

Zigbee Mobility Support

Some developments like Motorola’s neurRFon state that with the use of Zigbee it is possible to deploy applications with mobility support. This statement is true from the NWK point of view. Thus any device participating in a mesh network is able to find a route path and transmit data to any reachable (by multi-hop) destination. This is also valid if devices move, since route requests can be issued when a wireless node has moved out of range of its previous router. However this assumption is
only applicable to FFDs, thereby only routers and the coordinator are able to issue route requests and route replies. This fact rises a new problem since Zigbee v1.1 does not define low power modes for routers or coordinators. Moreover no location awareness services are supported neither intra-PAN or inter-PAN handovers are covered by the specification [32].

2.5 Chapter Summary

The fundamental properties and features of mobile WSNs have been briefly introduced. The overview covered the main applications and design challenges of WSNs and mobile communications. Moreover the architectural description of the sensors developed in this thesis has been provided. The Bluetooth stand-alone sensor architecture reduces the number of discrete components for embedded wireless sensors, however, the sampling frequency is reduced to 10 Hz. Furthermore Bluetooth and Zigbee have been presented investigating their benefits and limitations for implementing mobile WSNs. These limitations are expressed in terms of location awareness, handover and routing capabilities of both technologies. The performed investigations serve as background theory for the implementation of such capabilities in Chapters 3, 4, 5 and 6 respectively.
Chapter 3

Local Positioning with Bluetooth

Local positioning refers to the required techniques, methods and algorithms to discover the position of a certain object or person indoors. This concept is opposed to global positioning where position is determined outdoors like it occurs in the Global Positioning System (GPS). Self-organizing WSNs are one of the systems that would benefit from the new local positioning features offered by the new generation of wireless systems. Location dependent sensor data transfers could be optimized by means of local positioning services [54]. Local positioning for wireless sensors increases mobility management [48] quality and overall network data rate by means of minimizing ad-hoc routing paths. Demands for low power consumption, security and integrity are also increased by location dependent cell sizing. Many start-up companies have available proprietary systems meeting the unique requirements of each application. Therefore the use of standards like Bluetooth, WLAN IEEE Std 802.11 family and Zigbee is a step towards a universal solution.

This chapter starts investigating different positioning techniques and the viability for their adaptation to Bluetooth. Investigations and calculations of different Bluetooth standard parameters (RSSI and transmitted power) that may be used for distance estimation follow. Those parameters are then tested in a commercial platform from CSR. Moreover the algorithm for a tracking system using triangulation is provided. The last part of the chapter analyzes the accuracy of the algorithms referred to the initial theoretical results. The chapter ends with a summary discussing about the usability of the results.
3.1 Problem Definition

The chapter’s introduction has stated the needs and advantages of using wireless communication standards for implementing local positioning in WSNs. Some WSNs obtain their positioning information from a location standard like GPS, this implies a higher and more expensive sensor since two receivers are needed, one for communication and one for location. The challenge is then to reduce the size and cost of the sensor while still providing acceptable location accuracy. When Bluetooth is used, the main problem that the designer encounters is that Bluetooth v1.1 does not provide local positioning services [50]. However the local positioning working group of the Bluetooth SIG is putting effort in providing this functionality in future versions.

However a positioning system could be implemented with Bluetooth if there was a way of extracting location information from the communication’s front-end. Extracting location information from a Bluetooth’s front-end is not a common practice due to the short radius of most Bluetooth piconets. Therefore most of location systems offered with Bluetooth are based on Cell Identification (Cell-ID) [55]. The advantage of Cell-ID systems is their simplicity and reduction of location information acquiring time. The main drawback of Cell-ID implementations is the low accuracy obtained which is up to a piconet’s range.

Other works are based on signal strength measurements with resolutions down to 3.7 m [56] and 1.2 m [57]. These works are focused in a single Bluetooth parameter, the RSSI. Since measurements of RSSI are restricted by the use of TPC, alternative methods must be investigated.

3.2 Goal: Provide Local Positioning Services to Bluetooth

The main goals in this chapter are to investigate and develop a local positioning system with Bluetooth. This system will provide valuable information for the WSN mobility management entity. The mobility management entity can then use the local positioning service to aid other WSNs tasks as handover and ad-hoc routing.

The goal of this work is not to improve location accuracy of previous works since multi-path fading and other signal propagation phenomena affect the measurements. Moreover time related effects of positioning tasks discussed in other works [58], are out of the scope of this work.

Instead this work focuses on investigating the different alternatives that Bluetooth provides to implement positioning systems. Thereby the main goal is to provide theoretical calculations and experiment results of not only RSSI functionality but also from transmitted power. Thus the results of the work can be applied to all Bluetooth device classes.
3.3 Relevant Existing Positioning Techniques

In order to decide which positioning technique fits Bluetooth better, an investigation of existing techniques is required. The following sections present a short overview of some of the most relevant positioning techniques.

3.3.1 Carrier Phase

Carrier phase measurements can be used to measure distances [59]. Figure 3.1 illustrates how this technique works. The distance between two transceivers \(d\) is then determined by (3.1), where \(N\) is the number of periods, \(\lambda\) is the wavelength and \(\phi\) is the phase.

\[
d = N \lambda + \phi.
\]  

(3.1)

3.3.2 Radio Signal Strength

This technique measures the signal strength of a RF wave (transmitted or received). If measurements are taken at the receiver, the received strength of the signal is related to the remote sender’s location. If measurements are performed at the transmitter, it is possible to measure the needed local transmitting output power to reach a remote receiver at a certain distance. Distance is then estimated using propagation equations and models as Rappaport describes in ref. [44].

3.3.3 Time of Arrival (TOA)

The time that takes for a wave to arrive to a destination can be measured and thereafter the distance can be calculated if the wave’s propagation velocity is known.

3.3.4 Angle of Arrival (AOA)

With this technique it is possible to determine the position of a mobile device with only two landmarks as Fig. 3.2 illustrates. However this solution is not feasible for
3.3.5 Cell Identification (Cell-ID)

The position of a device is given by its cell’s position. Each cell has a cell BS with known position. This means that a sensor moving inside a cell will always have the same location given by its cell BS’s location. This is the simplest technique which may be useful in applications demanding low accuracy.

3.4 Distance Estimation with Bluetooth

From preceding section it can be concluded that Bluetooth should support mechanisms to measure phase, time, angle, energy or association to a cell (a master’s cover area) in order to be able to infer its position. Phase and angle measurements are not provided by the Bluetooth standard so they are discarded [51]. Time of flight measurements with a Bluetooth timing deviation of packet transmission up to 1 µs may result in unacceptable distance errors [56].

However the standard provides a mechanism to find out the received and transmitted power in a Bluetooth transceiver given by the RSSI and transmitted power parameters respectively [54]. Moreover Bluetooth supplies a mechanism to measure the Link Quality (LQ) of a link. These three last parameters provide information that can be converted into distance [60], [61]. Cell positioning is also supported if the physical area covered by a piconet is considered as a cell, [62], [63]. The accuracy of distance measurements ranges then from 1 m to 100 m if class 3 and class 1 devices are respectively used. Even though the accuracy obtained with class
3.4. DISTANCE ESTIMATION WITH BLUETOOTH

3 devices is 1 m, they are not used in Cell-ID systems due to the expenses of deploying a lot of cells to guarantee area coverage. Due to the low accuracy obtained with Cell-ID, this method will not be further investigated in this thesis.

3.4.1 Transmit Power Control (TPC)

As defined in section 2.3.1 Bluetooth provides three power classes where class 1 corresponds to the highest power up to 20 dBm or 100 mW. TPC refers to the mechanisms involved in the control of the output power of a class 1 Bluetooth device in order to save power and minimize interference [50]. Bluetooth defines a mandatory power control scheme for any Bluetooth device with a maximum transmitting power greater than 4 dBm. For classes 2 and 3 this feature is optional. In order to implement TPC it is required that the sender supports LM TPC commands and the receiver supports Golden Receiver Range (GR) and RSSI. When using TPC the remote receiver’s LM constantly monitors the RSSI. If the RSSI is above the GR, a decrease in the local output power is requested and a similar sequence occurs if the remote’s RSSI falls below the GR but requesting a local increase in the power.

3.4.2 Received Signal Strength Indicator (RSSI)

RSSI implementation is optional for all Bluetooth devices but most of commercial devices offer it. Moreover RSSI support is mandatory in newer versions of the specification. RSSI is used by a local device to control the level of the remote device’s output power. RSSI values may be read separately for each ACL connection over the HCI issuing the standard HCI Read RSSI command. The obtained value after issuing the HCI Read RSSI command is not an absolute measurement of the actual received signal strength but a relative indicator for power control purposes. The obtained RSSI value is referred to the limits of the GR since the local device must only know if the remote device is transmitting with too low or too high energy. The Bluetooth specification only states that a device should return a RSSI value greater than zero if the received power level is above the GR, a negative RSSI value when the received power level is below the GR and if the received power level is within the GR the returned value should be zero as Fig. 3.3 shows. This specification openness limits the use of RSSI as a distance estimation parameter since the specification does not define the RSSI readings accuracy and thereby manufacturers may provide specific levels of accuracy.

3.4.3 Golden Receiver Range (GR)

Bluetooth receivers have a 20 dB wide window for desired operation [5]. If the received power is outside this window because it is too high, the front-end might be overloaded. On the other hand if the received power is too low the BER may rise.
This power window is designated as GR by the Bluetooth specification. The lower limit value of the GR is defined by the Bluetooth specification and it is between -56 dBm and 6 dB above the receiver’s sensitivity for a particular implementation. The receiver’s sensitivity is defined as the input level for which a BER of 0.1 % is met and it shall be below or equal to -70 dBm. The GR’s upper limit value should be 20 dB $\pm$ 6 dB above the lower limit.

### 3.4.4 Transmitted Power

For distance measuring applications using transmitted power, the system must support TPC. If TPC is not supported the output power will be kept constant and no distance information may be inferred from its value. Local transmitted power may be read in a Bluetooth device issuing the HCI Read Transmit Power Level command [51]. The returned value is a number between -30 and 20. In this case the Bluetooth specification does define a range for the accuracy of the readings given by power steps between 2 dB and 8 dB [50]. Current specification does not allow Bluetooth devices to request a specific step size while performing TPC (although future revisions of the specification will include this functionality). Thus only a step increase or decrease can be requested. If the transmitter power level is equal to its maximum and a LMP incr_power_req message is received, the transmitter’s LM will answer back with a LMP max_power message as depicted in Fig. 3.4 d). A similar operation occurs if the transmitter’s power is at its minimum value and a lower power is requested by the receiver. Assuming a step size of 8 dB the theoretical distance measurement accuracy is illustrated in Fig. 3.5. The values are obtained using the Bluetooth propagation model described in section 3.5 with a path-loss coefficient equal to 2.7, and the golden range lower limit placed at -74 dBm. Output powers vary from -28 dBm to 20 dBm.
3.4.5 Link Quality (LQ)

Link quality can be obtained for a specific Bluetooth link, again the Bluetooth specification does not provide a deterministic definition of the parameter. LQ readings are obtained issuing the HCI _Read_Link_Quality_ command. The LQ reading should be between 0 and 255. The higher the value is, the better the link quality is. Bluetooth vendors decide how to measure LQ. LQ usually degrades if the distance between two transceivers increases, this metric is usually related to the BER. It may be possible then to infer the distance to a remote device using LQ.

3.5 Bluetooth Mobile Radio Propagation Model

Signal power information gathered with RSSI and transmitted power must be translated into distance information. If the distance between a transmitter and a receiver is called “path”, it can be stated that there are Path-Losses (PLs) that affect the strength of the signal. If only direct propagation (line of sight path) mechanism is considered the received power at a distance \(d\) can be calculated by the Friis free space equation [44] given by:

\[
P_r(d) = P_t G_t G_r \frac{\lambda^2}{(4\pi)^2 d^2 L}.
\]

Where \(P_t\) is the transmitted power, \(P_r\) is the received power, \(G_t\) is the transmitter antenna’s gain and \(G_r\) is the gain of the receiver antenna. The carrier wavelength
\( \lambda \) is 0.122 m at 2.45 GHz. \( L \) is the system loss factor not related to propagation. However (3.2) is restricted to the Fraunhoffer region, also called far-field region. The Fraunhoffer region starts when \( d > \lambda \) for half-wavelength antennas. Assuming no system losses, if (3.2) is expressed in dBm then (3.3) is obtained where \( n \) is the path-loss coefficient and \( P_r \) and \( P_t \) are the received and transmitted powers in decibels relative to one milliwatt [64].

\[
P_r(dBm) = -20 \log_{10} \left( \frac{4\pi}{\lambda} \right) - 10n \log_{10}(d) + G_t + G_r + P_t(dBm). \tag{3.3}
\]

Simplifying (3.3) and keeping the terms associated to the path between the transmitter and the receiver (3.4) is obtained. Where \( PL \) stands for Path-Loss and is given in dB.

\[
P_t(dBm) - P_r(dBm) = PL = 20 \log_{10} \left( \frac{4\pi}{\lambda} \right) + 10n \log_{10}(d). \tag{3.4}
\]

Since Bluetooth operates from 2.4 GHz to 2.4835 GHz the wavelength (\( \lambda \)) may be calculated at the center frequency of 2.442 GHz obtaining \( \lambda = 12.3 \) cm. Combining this value with (3.4) the simple expression of (3.5) is obtained.

\[
PL = 40.2 + 10n \log_{10}(d). \tag{3.5}
\]

Typical path-loss coefficient (n) values are presented in Table 3.1 [64].

---

Figure 3.5: Transmitted power vs distance with step size equal to 8 dB.
3.6 CSR Bluetooth Implementation

CSR BC2 Bluetooth modules have been used to collect the data presented in section 3.9. As stated before, some Bluetooth parameter attributes can be defined by the manufacturer, therefore it is interesting to investigate CSR’s particular implementation of TPC, RSSI and GR. The investigation is based on public documents from CSR and experimental results.

3.6.1 CSR TPC

In order to configure BC2 modules to support TPC it is necessary to configure the Persistent Store Keys (PSKEYs) of the module. PSKEYs are settings saved in the internal memory of the module that CSR may modify to change various characteristics of a Bluetooth transceiver. The LC\_PEER\_POWER\_PERIOD PSKEY must be configured with an amount of time specified in $\mu$s. This time is the period between attempts to change the peer’s transmit power. The power steps are defined in PSKEY LC\_POWER\_TABLE. This table includes the values of the gain for the internal Power Amplifier (PA), the external PA (if used), and the total output power in dBm. An example of a power table is provided in Table 3.2.

3.6.2 CSR RSSI

In order to extract location information from RSSI readings, TPC must first be disabled. This is done by setting the module’s LC\_PEER\_POWER\_PERIOD PSKEY to 0. Otherwise most of RSSI readings will be 0, which by definition is associated to a 20 dB band. Moreover the maximum transmitted power when TPC is not enabled should be also set using PSKEY LC\_MAX\_TX\_POWER. The Bluetooth specification defines a maximum output power value equal to 4 dBm for devices without TPC support.

RSSI readings are obtained issuing the HCI\_Read\_RSSI command which returns a signed 8-bit integer giving values between -128 and +127 under and over the GR as Fig. 3.6 illustrates for an ACL connection. In BC2 if the RSSI rises above the

---

Table 3.1: Path-loss coefficient values.

<table>
<thead>
<tr>
<th>Type of Clutter</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (Freespace)</td>
<td>2.0</td>
</tr>
<tr>
<td>Light</td>
<td>2.5</td>
</tr>
<tr>
<td>Moderate</td>
<td>3.0</td>
</tr>
<tr>
<td>Heavy</td>
<td>4.0</td>
</tr>
</tbody>
</table>
Table 3.2: Original power table from a BC2 CSR module.

<table>
<thead>
<tr>
<th>Internal PA</th>
<th>dBm</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>−24</td>
</tr>
<tr>
<td>14</td>
<td>−20</td>
</tr>
<tr>
<td>20</td>
<td>−16</td>
</tr>
<tr>
<td>27</td>
<td>−12</td>
</tr>
<tr>
<td>34</td>
<td>−8</td>
</tr>
<tr>
<td>42</td>
<td>−4</td>
</tr>
<tr>
<td>49</td>
<td>0</td>
</tr>
<tr>
<td>58</td>
<td>3</td>
</tr>
<tr>
<td>63</td>
<td>6</td>
</tr>
</tbody>
</table>

GR upper limit the RSSI will increase its value one step for each dB the signal is above the GR. The reading values will limit between 20 and 30. If the RSSI drops below the GR lower limit the reading will vary between -1 and -10 but it does not give any information about how far the incoming signal is below the GR’s lower limit.

3.6.3 CSR GR

CSR defines the GR by means of one PSKEY which sets the bottom level of the GR, the PSKEY is called LC_RSSI_GOLDEN_RANGE, with a default value of 80. The upper limit of the GR is automatically set to 20 dB over the lower limit.

3.6.4 CSR LQ

CSR bases the LQ BER measurements on Forward Error Correction (FEC) failures. LQ readings are obtained issuing the HCI_Get_Link_Quality command which returns a number that is directly equivalent to the BER between 255 (0 % BER) and 215 (0.1 % BER). LQ readings below a value of 215 are not fully characterized by CSR.

3.7 Distance Estimation Algorithm

The distance estimation algorithm is based on the Bluetooth mobile radio propagation model obtained from Friis free space equation given by (3.5). Thereby distance estimation is calculated with a known transmitting power ($P_t$), received power ($P_r$)
and the path-loss coefficient \( n \) as described in (3.6).

\[
d = 10 \frac{P_t - P_r - 40}{10^n}. \tag{3.6}
\]

It is important to remark that \( P_r \) is obtained from the RSSI value taking into account the GR limits as (3.7) and (3.8) describe where \( GR_U \) and \( GR_L \) are the upper and lower GR limits respectively. Moreover several RSSI measurements must be collected (between 3 and 5) and averaged before calculating \( P_r \) when measuring indoors. This is done to reduce signal strength variations due to signal fading.

\[
P_r(dBm) = RSSI + GR_U \quad \text{for} \quad RSSI > 0. \tag{3.7}
\]

\[
P_r(dBm) = RSSI - GR_L \quad \text{for} \quad RSSI < 0. \tag{3.8}
\]

On the other hand if transmitted power is used, the receiver’s GR lower limit must be known in advance. Then the algorithm can estimate the distance as the center point of the given range obtained for a certain output power as illustrated in Fig. 3.5.

### 3.8 Triangulation Algorithm

The triangulation algorithm is a GPS like algorithm illustrated in Fig. 3.7, this algorithm has been previously reviewed by D. Niculescu and B. Nath in ref. [65].
CHAPTER 3. LOCAL POSITIONING WITH BLUETOOTH

This algorithm needs the information of at least three landmarks ($r_i$) and their respective distance ($|p_i|$) to the real point. The distance $|p_i|$ is obtained as result of the estimated value obtained with the Bluetooth mobile radio propagation model described in the previous section. The algorithm needs an estimated point $\hat{r}_u$ before initiating. The algorithm then starts by calculating $|\hat{p}_i|$ and $|\Delta p|$:

$$|\hat{p}_i| = \sqrt{(r_{ix} - \hat{r}_{ux})^2 + (r_{iy} - \hat{r}_{uy})^2}. \quad (3.9)$$

$$|\Delta p| \approx |\hat{p}_i| - |p_i|. \quad (3.10)$$

Then the unit vector of $\hat{p}_i$ is given by:

$$\hat{I} = -\frac{r_i - \hat{r}_u}{|r_i - \hat{r}_u|}. \quad (3.11)$$

A new estimated point can be calculated with $(\Delta x, \Delta y)$ from the linear system $A=BX$ as described below.

$$\begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \Delta p_3 \end{bmatrix} = \begin{bmatrix} \hat{I}_{1x} & \hat{I}_{1y} \\ \hat{I}_{2x} & \hat{I}_{2y} \\ \hat{I}_{3x} & \hat{I}_{3y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}. \quad (3.12)$$
The algorithm then calculates the new estimated point with \((\Delta x, \Delta y)\) and starts over again until the error is much lower than the system resolution. Equation 3.12 was solved in Matlab, however a different approximation is used in the implemented program, to solve the system. The problem is solved in three different stages. This is done by splitting the system in three different subsystems and solving them separately as described by 3.13 - 3.15.

\[
\begin{bmatrix}
\Delta p_1 \\
\Delta p_2 
\end{bmatrix} = \begin{bmatrix}
\hat{I}_{1x} & \hat{I}_{1y} \\
\hat{I}_{2x} & \hat{I}_{2y}
\end{bmatrix} \begin{bmatrix}
\Delta x_1 \\
\Delta y_1
\end{bmatrix},
\]

(3.13)

\[
\begin{bmatrix}
\Delta p_1 \\
\Delta p_3 
\end{bmatrix} = \begin{bmatrix}
\hat{I}_{1x} & \hat{I}_{1y} \\
\hat{I}_{3x} & \hat{I}_{3y}
\end{bmatrix} \begin{bmatrix}
\Delta x_2 \\
\Delta y_2
\end{bmatrix},
\]

(3.14)

\[
\begin{bmatrix}
\Delta p_2 \\
\Delta p_3 
\end{bmatrix} = \begin{bmatrix}
\hat{I}_{2x} & \hat{I}_{2y} \\
\hat{I}_{3x} & \hat{I}_{3y}
\end{bmatrix} \begin{bmatrix}
\Delta x_3 \\
\Delta y_3
\end{bmatrix}.
\]

(3.15)

Calculating the mean value of the points from the equations above \((\Delta x_1, \Delta y_1)\), \((\Delta x_2, \Delta y_2)\), \((\Delta x_3, \Delta y_3)\):

\[
\Delta x = \frac{\Delta x_1 + \Delta x_2 + \Delta x_3}{3},
\]

(3.16)

\[
\Delta y = \frac{\Delta y_1 + \Delta y_2 + \Delta y_3}{3}.
\]

(3.17)

\((\Delta x, \Delta y)\) are used in (3.18) and (3.19) to calculate a new estimated point. The whole process is then iterated until the error is below 0.00001 m.

\[
\hat{r}_{ux} = \hat{r}_{ux} - \Delta x.
\]

(3.18)

\[
\hat{r}_{uy} = \hat{r}_{uy} - \Delta y.
\]

(3.19)

Both the approximation method and the original system were tested in Matlab to check the difference. The difference between the two solving methods is 0.000017 m for the x coordinate and 0.000032 m for the y coordinate. These errors are too small if compared with the system’s accuracy.

### 3.8.1 Self & Remote Positioning Topologies

Triangulation Bluetooth-based systems \[56\] employing signal power measurements to infer distances usually consist of three landmarks (access points), and a mobile Bluetooth-enabled sensor. The term landmark refers to a device in the WSN that knows its own position. Both APs and the sensor may work as measuring units and signal transmitters. The resulting systems have remote and self positioning topologies respectively \[66\]. When using remote positioning topologies each AP estimates the distance to a remote sensor. Distance information is then collected by a central location server, which calculates the sensor’s position. This topology
enables small mobile sensors, reducing cost and power consumption. On the other hand if the system uses self positioning topology, the sensor itself calculates its position from the estimated distance to the three APs. The tests in this thesis are based on a self positioning topology which is illustrated in Fig. 3.8.

### 3.8.2 Bluetooth Topology

Independently of the positioning topology the Bluetooth topology must be taken into account since it affects the QoS offered by the system. Considering the roles that a Bluetooth-enabled sensor can adopt, two different Bluetooth topologies are feasible to implement positioning systems as described below.

- **Bluetooth Sensor as Slave**
  APs are usually configured as Bluetooth masters in order to provide higher QoS in piconets including up to seven active slaves. The mobile wireless sensor is then configured as a slave. One drawback of this Bluetooth topology is that if tracking is demanded by the system, the sensor should establish three active ACL links simultaneously, thus forming a scatternet formed by three landmark piconets which will reduce the QoS offered by the sensor as described in section 2.3.3. In the worst case the sensor will reduce its QoS to the half, due to the guard slots needed to re-synchronize to the different landmark’s CLKNs and FHSs. Another drawback is that only seven active slaves can be tracked simultaneously. The advantage of this topology is the easiness for integration of remote positioning systems, thereby freeing the sensor from positioning management overheads. However the sensor slave must support simultaneous connections to three masters.
• **Bluetooth Sensor as Master**  
  When the mobile sensor is configured as a master, the resulting topology is a single piconet reducing the latency for location measurements and improving the QoS offered by the sensor. The major drawback of this topology is that a piconet is required for each sensor in the area and APs must be involved simultaneously in several piconets, thereby reducing the overall QoS.

The performed experiments in this thesis are based on a piconet where the sensor is configured with the role of master and the three landmarks are configured as slaves.

### 3.9 Measurements

Measurements were focused on RSSI, transmitted power and LQ to estimate distance. Triangulation estimation results are also reported at the end of this section. The measurements were performed indoors with direct line of sight between the transceivers. Distance measurements were performed by varying the distance between two BC2 modules and recording RSSI, transmitted power and LQ values versus distance.
3.9.1 Measurements Setup

Measurements were performed with one CSR’s Casira development board and an own-designed Casira-like board. Both boards included CSR BC2 modules. One of the boards was configured to work as a master and the other was configured as a slave. The slave was static and configured to accept connections running an embedded application. The master board was mobile and connected to a PC via serial port. The PC runs an application described in Fig. 3.9. The PC application was written in C implementing a graphical user interface using National Instruments (NI) Labwindows/CVI. The whole Bluetooth stack is embedded in CSR’s BC2. RS232 is used to communicate BC2 and the computer at APP level (not at HCI level), support for RSSI and power measurements is implemented at Device Manager (DM) level as Fig. 3.9 depicts. The user controlling the application in the master can issue RSSI, power or LQ read commands from the graphical interface and log the measurements to a file. All experiments were performed indoors with direct line of sight between the master and the slave, both antennas were oriented to achieve maximum transfer of energy. Each measurement was performed one hundred times.

3.9.2 Measurements Procedures

Three types of measurements were performed using the setup described in the previous section, corresponding to RSSI, transmitted power, and LQ. All measurements
followed the same procedure. Firstly, the PSKEYs must be set to configure TPC and maximum transmit power, then an ACL connection is established and the master is displaced while performing measurements.

3.9.3 RSSI Measurements

RSSI measurements report useful information when TPC is disabled by setting PSKEY LC_PEERPOWER_PERIOD to 0 in the mobile device (the master in this specific application) as described in section 3.6.2. By doing this, the master will not issue increasing or decreasing power commands to the slave’s LM. Thereby the output power of the slave will remain constant and signal attenuations will mainly be due to path-losses. If longer measurement ranges are required then the output power of the slave must be set to its maximum value which is +6 dBm in the case of CSR BC2 without an external PA, this is done by setting PSKEY LC_DEFAULT_TX_POWER and PSKEY LC_MAX_TX_POWER to 6.

RSSI measurements are not linear and they are not useful for values inside the GR. The worst case RSSI readings versus distance are illustrated in Fig. 3.10, the theoretical values obtained with the Bluetooth mobile radio propagation model are also plotted in the same figure. The path-loss coefficient value for the calculations (n) is 2.7 corresponding to light path-losses, the GR upper limit is set to -60 dBm and the output power to 6 dBm.

An average of all performed measurements reports better results as Fig. 3.11 shows. Since the GR is limiting the range of measurements it might be interesting
to modify the position and width of the GR to be able to extend the measurement range. This could be done by modifying PSKEYs related to RSSI such as LC_RSSI_GOLDEN_RANGE and attenuators settings. However, modifying these PSKEY values may result in operation failure of the front-end. Thus the modification of GR related PSKEYS was kept out of the scope of performed experiments.

3.9.4 LQ Measurements

The setup for LQ measurements is the same that was used for RSSI measurements. An average of the results of all tests is presented in Fig. 3.12. Obtained results are not valuable for distance estimation since the LQ value is constant (255) for a distance interval of 8 m.

3.9.5 Transmitted Power Measurements

Transmitted power measurements are once more performed with the same test scenario described for RSSI measurements. There are two PSKEYs that have been modified under the tests. The first one is LC_PEER_POWER_PERIOD which is configured with a value corresponding to the period time in $\mu$s of TPC updates.

BC2 modules are defined as class 2 devices but since their maximum output transmitter power (6 dBm) is over 4 dBm they must implement TPC. The second involved PSKEY is LC_POWER_TABLE as described in section 3.6.1. The first
performed test illustrated in Fig. 3.13 shows the average of transmitted power readings using the default power table given by Table 3.2. The theoretical transmitted power steps are also reported in the same figure. Theoretical values have been obtained for a lower range GR limit of -74 dBm and a path-loss coefficient equal to 2.7. The second performed test uses a modified power table which has power steps of 2 dB ranging from -22 dBm to 6 dBm, with the goal of reducing the distance step width at higher powers. Worst case results of the second test are presented in Fig. 3.14. Figure 3.14 also provides information about the asymmetric behavior obtained when measurements are taken getting away and getting closer from/to the static slave.

### 3.9.6 Triangulation Measurements

Triangulation measurements were conducted in order to test the viability of the algorithm to find converging solutions. The system could not be tested as a tracking system in real time due to antenna behavior. The antenna provided with BC2 is a dipole with high directivity. With this type of antenna a mobile sensor placed at equal distance from the landmarks will registry different readings from each landmark and distance estimations will not provide the accuracy described in next section. Errors of 10 dB where obtained depending on the antenna orientation for a given constant distance. This problem may be overcome if an isotropic antenna is used. However the triangulation system was tested by rotating the mobile sensor while keeping its position, to point a single landmark at a time and then registering
the reading. When all readings were collected, the triangulation algorithm was executed.

Obtained results proved the validity of the triangulation algorithm finding converging solutions for the position of Bluetooth-enabled sensors. Distance estimation measurements were averaged before processing then in the algorithm.

3.9.7 Distance Estimation Accuracy

The accuracy of the distance estimation algorithm is determined comparing the theoretical values obtained with the Bluetooth mobile radio propagation model, with the worst case measurements. It should be noticed that in order to properly characterize the Bluetooth modules, measurements should have been done by wiring both Bluetooth front-ends and inserting an attenuator.

- The accuracy for RSSI based measurements for our scenario, has a minimum of 3 m (inside a range of 10 m). However most of experiment results provided less than 1 m errors for a range of 10 m.

- The accuracy of distance estimations using transmitted power is lower due to the asymmetric behavior illustrated in Fig. 3.14. The obtained accuracy is up to 9 meters for a 20 m range. However most of the experiments exhibited an accuracy of 3 m in a range of 20 m.
3.10 Chapter Summary

Although it may be possible to implement tracking systems with Bluetooth using signal power measurements, the obtained resolution is highly dependent on the specific implementation of each vendor. This is due to the fact that the specification is very open in terms of defining accuracy for RSSI, GR and transmitted power as described in section 3.4. The experiments results of this chapter must be thereby considered as indications and not general design references since they have been obtained from the specific CSR implementation of Bluetooth.

However, the results are still useful from an application point of view. Distance estimations with transmitted power can be performed for standard class 1 devices, while RSSI based estimations are applicable to class 2 and 3 devices. An important observation is that it is not possible to implement both RSSI and transmitted power measurements simultaneously in the same transceiver. This is due to the fact that when TPC is enabled then most of RSSI measurements will report a value of 0. A practical implementation of APs supporting both RSSI and transmitted power measurements would be to use the gathered information by an inquiry to find out the sensor’s Bluetooth class, and thereby determine if TPC should be enabled. Unfortunately this can not be done for each Bluetooth link, i.e., once an AP has enabled/disabled TPC it will affect all sensors in its cover area.

The advantage of using RSSI as a distance estimation parameter is that it does not present the asymmetric behavior which is present with transmitted power based measurements. The drawback of RSSI readings is that the effective measurement range in meters is shorter than the one achieved with transmitted power readings. Moreover LQ readings present a very short measurement range in meters which make them not appropriate to be used as a stand alone parameter for distance estimation. Instead LQ measurements can be combined with RSSI and transmitted power to deliver a more accurate result.

The estimated distance can be used as a ”getting out of range indicator” which is used in Chapter 4 to assist handover decisions. Moreover simple tracking systems could be integrated in automatic doors detecting when sensors are moving towards or away from the door.

The performed investigations and experiments cover a wide range of theoretical and practical aspects that must be taken into account when deploying a positioning system with Bluetooth.
Chapter 4

Handover with Bluetooth

Bluetooth piconets can be considered as cells. In such scenario, cells are deployed over a spatial area where wireless sensors will occasionally cross a cell boundary. If that happens, the ideal situation is to transfer the current link to the AP of the new cell in a procedure known as a handover. If this handover does not take place, the link is lost. This behavior is similar to the one described in first and second generation mobile phone systems [67]. Handover is usually initiated based on the received signal strength. In this sense, results obtained for local positioning based on signal strength in Chapter 3 will be of use in handover. Handover decisions are related to the selection of the new AP and the establishment of new links with the goal of maintain QoS, this in turn is related to the problem of how to optimize the overall network topology when a sensor has changed its position. The main challenge of using Bluetooth to implement handover is how to provide this functionality keeping in mind that the specification does not provide support for such procedure [50].

This chapter investigates handover for Bluetooth-based mobile sensors. The chapter starts defining the challenges presented by Bluetooth for handover implementation and the importance of avoiding dropped links for sensing applications. Handover strategies are then investigated together with a viability study for implementation with Bluetooth. The chapter continues describing the proposed algorithms for different types of handovers with Bluetooth. Finally, test and measurements are provided with focus on handover latencies and effects on QoS. The analysis of measurements compares obtained results with theoretical calculations. The chapter ends with a summary highlighting the applicability of achieved results.
4.1 Problem Definition

Assuming that the deployed network is based on several APs (base stations) like e.g., inside hospitals, seamless connections may be achieved with handovers. This aspect is easily solved in mobile technologies such as GSM, however it is not a straightforward step with Bluetooth since the specification does not provide support for such functionality. The required handover algorithm must ensure that when a sensor node gets out of range from an AP or another relaying sensor node, it should join to a new node to continue being an active member of the sensor network. In resume, the moving node must perform the following operations, not necessarily in the specified order, if handover is to be achieved: stop data transmissions and disconnect from the previous node, search for a new node, synchronize and establish a new link to continue data transmissions. Since searching (discovering) and connecting (synchronizing) processes usually require a long time, this solution will restrict continuous data streaming from a mobile sensor node. This is a critical issue in real time applications where no data must be lost and quality of service must be kept within acceptable levels. A solution could be to store data while handover is performed but this is not a cost effective solution as mobile sensors will be more expensive due to RAM memory costs.

Moreover related works commonly focus on a single handover strategy [68], [69], [10] and results are presented in form of simulations [70], [71], [72]. Thereby, theoretical investigations of realizable Bluetooth handovers are needed together with the required validation of the theory by means of experimentation.

4.2 Goal: Provide Handover Capabilities to Bluetooth

There are two main goals in this chapter, the first one is to provide a variety of handover algorithms for Bluetooth together with their related theoretical characterizations. Thereby obtained results may aid the designer when deciding which algorithm is best suited for its specific application. The algorithms should focus in overcoming the two main problems presented when handing links between base stations: reduction of the handover latency and maintain system QoS. Reduction of handover latency will impact the sensor’s cost in terms of RAM requirements.

The second goal is to provide experimental results to analyze how real implementations cope with the proposed algorithms. Thereby practical conclusions may be extracted offering completing information for already existing simulation results.
4.3 Existing Handover Strategies, Types and Algorithms

Handover comprises an initiation phase and an execution phase. The initiation phase is usually based on a decision making strategy. This decision strategy is in its turn based on signal strength levels. The definition of these levels is dependent of the implemented algorithm. The execution phase includes allocation of new radio resources (e.g., synchronization and link establishment).

- Handover Strategies
  Handover operations are often managed by central mobility centers which in mobile telephony are called Mobile Switch Centers (MSCs) [48]. Mobility centers are usually hardwired to APs and collect data concerning channel occupancy status as Fig. 4.1 illustrates. In these type of infrastructure topologies both handover initiation and execution are overseen by the mobility center. Since a Bluetooth infrastructure network is composed by several APs where all of them are connected to the same mobility center it can be concluded that only intra-switch handovers will occur. Thereby, there is only one MSC overseeing all handovers. Depending on the used information and the action taken to initiate handovers, the strategies can be:
- Mobile Controlled Handover (MCHO)
  With this type of handover the mobility management overhead in the mobility center is reduced. On the other hand the mobile sensor node increases its complexity in order to perform handover initiation and execution.

- Network Controlled Handover (NCHO)
  In this strategy the APs monitor the signal strength from the mobile sensors and report the measurements to the mobility center. The mobility center is then fully responsible for choosing the next AP and initiating the handover. The mobile sensor has a passive role during the whole process.

- Mobile Assisted Handover (MAHO)
  This strategy is a mix of the two previously described strategies. In this case, the mobile sensor collects readings of different APs and report them back to the mobility center. The handover initialization relies on the mobility center.

- Handover Types
  As described in section 2.2.3 the handover types can be classified as soft and hard, however there are two more types known as backward and forward [48], a short review of all types is provided below:

  - Hard Handover
    This handover type is used in systems were mobile sensors only allow one connection at a time. This type of handover is also known as "break before make".

  - Soft Handover
    In soft handovers the mobile sensor supports more than one link to different APs simultaneously. This type is also known as "make before break".

  - Backward Handover
    This type of handover is characterized by a prediction ahead of the handover via the existing link. This type may present problems in the case of sudden losses.

  - Forward Handover
    In this type, the handover is initiated by the new radio link established with the candidate AP. This handover presents advantages in case of old link deterioration.

- Handover Algorithms
  There exist many handover algorithms [67]. The most common approach is to define a threshold level (T) at a predefined level above the front-end’s
4.4 HANDOVER SUPPORT IN BLUETOOTH

This algorithm is described by (4.1) where $\delta_{th}$ is the margin above the sensitivity of the radio and it usually takes values between 0 dB and 12 dB for first generation mobile systems and 6 dB for second generation mobile systems.

$$T(dBm) = \text{Sensitivity} + \delta_{th} (dB).$$  \hspace{1cm} (4.1)

With this algorithm the handover initiation takes place when the received power level is below the threshold value ($T$). However this algorithm involves several trade-offs:

- If $\delta_{th}$ is too small, too many dropped links will take place due to signal dropping below the sensitivity before the handover is finished.
- If $\delta_{th}$ is too large there will be too many handovers which will increase mobility management tasks reducing QoS in the system.
- Received signal strength level must be averaged to avoid momentary fading, this averaging task introduces a delay in processing.

It is important to note that handover frequency is inversely proportional to cell size and directly proportional to mobile sensor’s velocity. A conclusion of this section is that TDMA/FDMA systems (like Bluetooth) demand changes of frequencies and synchronization to the new AP which lead at first instance to hard handovers.

4.4 Handover Support in Bluetooth

Bluetooth does not currently support handover between piconets [50]. If a slave sensor is losing its link to one master, the specification does not make provision to transfer the link to a new master, thus disconnections occur if slaves get out of range from a master.

As described in Chapter 2 Bluetooth supports two network topologies referred to as piconets and scatternets. Since a slave needs to always be connected to one master, intra-piconet handover is not possible. On the other hand in scatternets there are two or more masters present. In this case it is possible to perform handover if the network supports ad-hoc routing based on BDAs. Another scenario where handover may take place is the typical infrastructure topology described in Fig. 4.1 where a mobile sensor may perform inter-piconet handovers. However the specification does not cover any of these possible scenarios as the specification just covers the design of a short-range radio link, not the networks on top of it. Some of the reasons why Bluetooth does not support handover are:

- Bluetooth is originally intended for cable replacements and ah-hoc networks but not for mobile networks or WLANs.
- Bluetooth is intended to be a low-cost standard, it is supposed to be simple and small. Handover support will increase the complexity of the units and thereby their cost.
• In a Bluetooth ad-hoc network every Bluetooth device may take the role of master or slave in a dynamic fashion, thus it is hard to define the roaming boundary. Handover must be performed fast to follow this dynamic environment, since discovering (inquiry) and connecting (paging) processes require a long time (often between two and ten seconds), handover latencies are high.

However the investigations and experimental work performed in this thesis as well as related works [10] are conducted to overcome all these drawbacks. Thereby one of the main challenges is to develop a handover solution which keeps inter-operability and is vendor independent.

4.4.1 Bluetooth Parameters for Handover Algorithm Implementation

Bluetooth provides several parameters that may be used in conjunction with (4.1) to implement a handover algorithm. These parameters and their implications have been described in section 3.4. RSSI readings are directly compared with the threshold level (T) calculated in (4.1). For transmitted power readings (4.1) must be converted to:

\[
T(dBm) = \text{Minimum Transmitted Power} + \delta_{th} (dB). \tag{4.2}
\]

4.4.2 Time Analysis of Bluetooth Connections

As described in the previous sections handover must be quickly performed to follow Bluetooth dynamic environment. As a consequence, the establishment of new Bluetooth link must be performed fast. Particularly link establishment is the most time consuming task in a Bluetooth handover, thereby it is worthy to investigate how Bluetooth links are established.

Before communication can commence between two Bluetooth devices, transmitters and receivers need to be synchronized as described in section 2.3.3. Consequently connection times can be lengthy. When creating a Bluetooth connection two delays are involved. The first one is the time needed to discover other devices in the neighborhood. This discovering method is referred to as "inquiry" by the Bluetooth specification. Inquiry may generally take up to ten seconds to find all the devices in the neighborhood. The second delay occurs when the link is established, the establishment task is referred to as "paging" by the Bluetooth specification. This procedure may take up to 2.56 s. It can be concluded that link establishment time is variable in Bluetooth. The theoretical values given in Table 4.1 are based on the time taken to complete a successful inquiry and page operations respectively. The minimum inquiry time is obtained when a slave receives the inquiry message from a master in its first scanning slot at frequency f(k) and a response follows in the next slot with the FHS packet. Thereby the total minimum time is equal to two slots time. The average inquiry time is obtained if the inquiry device gets enough
responses, however this value does not guarantee that all devices within range will be found. The maximum value is defined as a timeout set between 10.24 s and 30.72 s [5].

The minimum paging time is calculated like in the case of inquiry by assuming that the slave receives the page message in its first scanning slot. The total procedure requires then four slots which is equal to 2.5 ms. The average paging time is obtained assuming mandatory paging scheme (page mode R1) [5] and the maximum paging time is 2.56 s.

### Paging and Clock Offset

All Bluetooth devices have an internal native clock referred to as CLKN which ticks every 312.5 µs as described in section 2.3.3. A master paging device connecting to a slave should use an estimate of the CLKN of the slave page-scanning device, called CLKE. This estimated clock is calculated by (4.3) where $\text{Offset}$ is the difference in time units between the slave’s native clock and the master’s native clock. If the CLKN of the slave page-scanning device is known, the pager might be able to speed up the connection establishment down to 4 ms [5]. The native clock of a slave can be known in advance if a previous inquiry response has been received from that particular slave. Moreover there are other methods for communicating this information to a master as following sections explain.

\[
\text{CLKE} = \text{CLKN} + \text{Offset}.\tag{4.3}
\]

#### 4.4.3 Bluetooth Scatternets and Handovers

Devices participating in a scatternet may switch between piconets as described in section 2.3.3. Every time the device switches, communications are interrupted during a maximum time guard of two slots that is required to synchronize to the new time base [51]. This guard time will affect the performance of soft handovers implemented with Bluetooth.
4.4.4 ACL and SCO Links Interaction with Inquiry and Page

The impact of executing inquiry-scans while an ACL link is active has a negative effect in the throughput. This is due to the fact that inquiry and page operations take precedence over ACL links [5]. If for example the Generic Access Profile (GAP) is followed, then the inquiry-scanning device will enter inquiry-scan for at least 10.625 ms every 2.56 s. During these 10.625 ms (17 slots) no ACL data communications can be carried out, i.e., the inquiry-scan procedure uses up bandwidth which limits the data throughput of the ACL link. Moreover page-scanning must also be performed to allow incoming connections, the scan window must be at least 11.25 ms [5]. Since sensor nodes do not know when other device will connect to them, they must periodically scan for inquiries and page procedures reducing their data throughput. If QoS should be maintained in the application, then SCO links must be implemented since SCO links take priority over both inquiry and scan operations [5]. However this solution will reduce the probability of success for inquiries as described by J. Bray in ref. [5]. The implementation of wireless sensor nodes based on SCO links is out of the scope of this thesis. Only ACL links are used for sensor data transmissions therefore handover with SCO links is not covered in this chapter. From a power consumption perspective it is not optimal to place a sensor node in repetitive inquiry and page scan modes since the receiver will then be on for long periods thereby draining the battery of the sensor. Moreover sensor nodes will then allow any other sensor within range to find them, generating unnecessary inquiry responses. If the inquiring sensor tries to connect, it will then waste even more power on the sensor [5].

4.5 Design of Bluetooth Hard Handover Algorithms

Hard handovers imply that mobile sensors have only one active link at a time so a disconnection must precede any new connection (break before make) as described in section 4.3. Applied to Bluetooth, this means that the link between the actual master and the slave is broken before a new link is established with another master or slave. Depending on the mobile sensor Bluetooth role, different handover strategies are allowed as the following sections describe.

4.5.1 Algorithm for Hard Handovers with Mobile Master Sensor

Since a master always starts a connection, the required network topology illustrated in Fig. 4.2 is formed by only one piconet. This topology requires a MCHO strategy. The master is then in charge of monitoring the link quality and deciding when to initiate the handover [52]. Execution of the handover is also implemented by the master. The slave APs are totally passive in this form of handover and they are configured to response to inquiry requests and accept connection requests. The
handover algorithm is illustrated in Fig. 4.3. The decision for handover initiation is taken using (4.1) and (4.2) depending if RSSI or transmitted power readings are used respectively. The term averaged in Fig. 4.3 refers to the procedure of averaging the signal power readings. Signal power readings are obtained with a periodicity equal to timer overrun period and they are averaged three times before taking a decision. $T$ in Fig. 4.3 refers to the threshold level defined in (4.1) and (4.2) and is set to zero if working with RSSI readings as (4.5) defines. When RSSI readings are less than zero it means that the signal strength is under the lower limit of the GR. In other words, $T$ has a value given by (4.4) according to GR limit definitions of section 3.4.3.

$$T(dBm) = \text{Sensitivity} + 6\ (dB). \quad (4.4)$$

$$T(RSSI) = 0. \quad (4.5)$$

The main advantage of this algorithm is that link establishment is fully compatible with the standard. However, the drawback of this algorithm is the high handover latency given by (4.6) assuming average inquiry and page durations. The handover latency ($t$) is defined as the time between the last received data in the first AP and the first received data in the new AP.

$$t = 2 \cdot \text{slots} + \text{Inquiry Time} + \text{Page Time} = 6.28\ s. \quad (4.6)$$

The first two slots are required to send the HCI Disconnect command and receive an ARQ baseband ACK.
4.5.2 Algorithm for Hard Handovers with Mobile Slave Sensor

If the mobile sensor is a slave then APs will be configured as masters, thereby there are several piconets in the area as illustrated in Fig. 4.4. This topology requires a NCHO strategy since connections are started from APs. APs are then in charge of monitoring the link quality and deciding when to initiate the handover. Execution of the handover is implemented by the mobility switch center. The mobile sensor is passive during the handover, slaves must be configured to perform both inquiry-scan and page-scan while not connected to an AP. The handover algorithm is illustrated in Fig. 4.5. Once more the decision for handover initiation is taken at the AP using equations (4.1) or (4.2) depending if RSSI or transmitted power readings are used respectively. The chosen candidate AP by the MSC is elected based on positioning information or by another type of algorithm. If local positioning is not available then the MSC chooses several APs in the vicinity of the first AP. The two main advantages of this algorithm is that it is fully compatible with the standard and that inquiry is no longer needed since the slave’s BDA is known and passed over
the backbone network (e.g., ethernet). Handover latency is given by:

\[ t = 2 \cdot \text{slots} + \text{Page Time} = 1.28 \text{ s}. \]  

(4.7)

The advantage over MCHO hard handover algorithm is that inquiry-scan is only executed once (when the mobile sensor first joins the network). However data throughput in piconets established by APs will decrease while APs perform scan procedures as scanning procedures take precedence over ACL links as described in section 4.4.4. This means that a master AP will interrupt its ACL activity while a connection is being performed, this time could be up to 2.56 s as described in Table 4.1.

Upgrade for Bluetooth NCHO Hard Handovers with Slave Sensors

The upgrade consists in reducing paging time by passing the slave’s clock offset value between APs. As described in section 4.4.2, connection establishment time may be reduced down to 4 ms if the page-scanning device’s (the mobile slave in this case) CLKN is known ahead. Before executing handover, the current AP issues a HCI\_Read\_Clock\_Offset command, this value is then transferred via the backbone network to the MSC which in turns delivers it to the chosen alternative AP. If the backbone network is used to implement a general clock for all APs the problem is solved as described above. If no shared clock is implemented then the current AP and the chosen alternative AP should execute a page procedure to find out their clock offset. This operation may be done at network deployment time. The clock offset to be used by the alternative AP is given by:

\[ \text{Offset}_{\text{AP2-Sensor}} = \text{Offset}_{\text{Sensor-AP1}} - \text{Offset}_{\text{AP1-AP2}}. \]  

(4.8)
Discussion

When mobile sensors are configured as slaves and the MSC is in charge of starting inquiry and page procedures, ad-hoc networking is not supported since mobile slave sensors are not able to join the network by themselves. This is due to the fact that a connection can only be initiated by a master AP. This rises the problem of how to support network scalability. I.e., how to expand the network by adding new sensors. This is solved by the Bluetooth specification in profiles such as Local Area Network (LAN) access where APs are discoverable (accept inquiries) and connectable (accept page) while being masters of their respective piconets. In this case a mobile slave sensor can initiate a connection by becoming a master (referred to the AP) for a short while. A role switch must then follow so that the AP becomes master again and continues to maintain communications with its slaves [5]. However, this procedure reduces the throughput of the system due to the priority of inquiry and page scans over ACL traffic. ACL data packet losses may occur when the sensor’s host (external processor executing the upper layers of the stack) receives a disconnection complete event. The Bluetooth specification defines the disconnection behavior assuming that all unacknowledged HCI data packets that has previously been sent to the Bluetooth controller (Bluetooth module implementing lower layers of the stack) have been flushed. In other words, all data packets stored in the...
4.6. Design of Bluetooth Soft Handover Algorithms

When a mobile sensor is connected to more than one AP simultaneously while executing a handover, the handover is of a soft type as described in section 4.3. In soft handovers the mobile sensor connects to an alternative AP before breaking the current link. Within Bluetooth terminology, it can be expressed as a mobile slave or master that participates in the creation of a new ACL link while keeping the old ACL link with another master or slave. Only after data sources and sinks have been redirected the first ACL link is dropped. Depending on the Bluetooth role assumed by the mobile sensor the following handover strategies are allowed.

4.6.1 Algorithm for Soft Handovers with Mobile Master Sensor

The network topology deployed to implement this algorithm is illustrated in Fig. 4.6. The main difference from hard handover topologies is that connections are point (master) to multi-point (slaves) for one piconet. Since handover initiation and execution relies on the mobile sensor node the required handover strategy is MCHO [8]. The mobile master sensor monitors the signal strength and decides when to initiate the handover following the algorithm illustrated in Fig. 4.7.
are passive during handover and they are configured to perform inquiry and page scans while being connected to mobile sensors. The decision for handover initiation is taken using (4.1) or (4.2) depending on if RSSI or transmitted power readings are used respectively as described in section 4.5.1. This algorithm is based on standard links. Due to inquiry and page procedures the sensor will anyhow stop data transmissions to the first AP for 6.28 s assuming average values for inquiry and page.

The advantage of this algorithm is that if the mobile sensor changes its direction while performing handover, it will not need to reestablish a link with the first access point. Thereby this algorithm is more reliable than its hard version.

Upgrade for Bluetooth MCHO Soft Handovers with Master Sensors

A variation of this algorithm is presented in Fig. 4.8 where the inquiry stage is no longer needed (handover latency = 1.28 s). Since APs are connected via the backbone network, the BDA of the new AP could be designated by the MSC and transmitted from the previous AP. This implies network activity during handover initiation thus the resulting handover strategy is MAHO.
4.6. DESIGN OF BLUETOOTH SOFT HANOVER ALGORITHMS

Figure 4.8: MAHO soft handover with mobile master sensor.

4.6.2 Algorithm for Soft Handovers with Mobile Slave Sensor

In this type of handover the mobile slave sensor is connected simultaneously to two APs. The Bluetooth network topology is a scatternet that will last until the handover is finished. The network topology deployed to implement this algorithm is illustrated Fig. 4.9. The handover strategy is MAHO since the network is involved in handover initiation and both network and mobile sensor are involved in handover execution [73]. APs monitor the signal strength of the link and decide when to initiate the handover by sending a message to the MSC. The MSC will then choose one or several APs that will attempt to connect to the mobile sensor. Once the sensor is connected to a new AP it will switch data streams from the previous AP to the newest. As soon as the new AP receives data from the mobile sensor it will send a message to the MSC, which will in turn notify the first AP to disconnect from the mobile sensor as illustrated in Fig. 4.10. This algorithm is designed to minimize handover operations in the mobile sensor. To avoid packet losses when disconnecting, the disconnect command from the first AP should be issued after the sensor has started to transmit to the new AP. The main advantage of this algorithm is the theoretical low latency which is reduced to the time needed to establish an ACL connection up to LM level. This time corresponds to the procedures illustrated
CHAPTER 4. HANDOVER WITH BLUETOOTH

Figure 4.9: MAHO soft handover topology with mobile slave sensor.

in Fig. 4.11. Thereby the handover latency for an ACL connection with exchange of features is given by:

\[ t = 4 \text{ slots (BB page)} + 8 \text{ slot (LMP)} = 7.5 \text{ ms.} \]  

(4.9)

However the system’s throughput is lowered during the handover:

- Regarding APs
  The impact of executing page while ACL links are active with mobile sensors has a negative effect in the throughput. During a time of up to 2.56 s, the AP is not able to receive data from its already active links. Clock offset information must be used to minimize this paging time. Mobile sensors should perform data flow control and save data in buffers while their associated AP is involved in a handover.

- Regarding Mobile Slave Sensor
  A mobile slave sensor which periodically performs page-scans will lower its data rate as discussed in section 4.4.4.

Upgrade for Bluetooth MAHO Soft Handovers with Slave Sensors

Infrastructure topologies like LAN based networks demand APs to be configured as masters to optimize network throughput. However paging procedures are time consuming and during paging an AP can not offer service to the already connected mobile sensors. Bluetooth offers the possibility to implement master/slave role switching. Once a link is established any device can request a switch in roles with respect to another device [50]. Thus a master AP in an existing piconet allows to be
4.7 Resume of Bluetooth Handover Algorithms

Table 4.2 presents a resume with the differences of previous discussed algorithms. This table has been filled with theoretical throughput and latencies assuming GAP recommendations executing inquiry-scans during 10.625 ms every 2.56 s. Assuming that page-scan mode is set to R2 then the device scans during at least 11.25 ms every
2.56 s. Moreover it has also been assumed DH5 packet type with symmetric links, and two guard slots for scatternet synchronization as described in section 2.3.3. The theoretical values for the different algorithm types are:

- **Bluetooth Hard Handover**
  If an AP is a slave it is then involved in a scatternet while performing inquiry and page scans respectively. The throughput of the AP is given by (4.10). In (4.10) \( S \) refers to Slots, \( S_i \) refers to Slots needed for inquiry-scan, \( S_p \) refers to Slots needed for page-scan, \( B \) refers to bytes and \( b \) to bits.

\[
\text{Throughput}(AP_{i+p}) = \frac{(S_{2.56} - S_i - S_p)(339B)}{12 S} = 358.068 \text{ Kbps}. \quad (4.10)
\]

If the BDA is known ahead then inquiry is not performed and the throughput of the slave AP is given by:

\[
\text{Throughput}(AP_p) = \frac{(S_{2.56} - S_p)(339B)}{2.56 S} = 359.128 \text{ Kbps}. \quad (4.11)
\]

Assuming an average connection time including inquiry and page of 6.28 s a mobile master sensor must buffer its data during this time since this time corresponds to the time the mobile sensor is not connected to any AP.
Table 4.2: Bluetooth handover algorithms resume.

<table>
<thead>
<tr>
<th>Handover Type</th>
<th>AP Role</th>
<th>Sensor Role</th>
<th>Latency (s)</th>
<th>Throughput AP (Kbps)</th>
<th>Throughput Sensor (Kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard(I + P)</td>
<td>S</td>
<td>M</td>
<td>6.281</td>
<td>358.068</td>
<td>Buffer 6.281 s</td>
</tr>
<tr>
<td>Hard(I + P)</td>
<td>M</td>
<td>S</td>
<td>6.281</td>
<td>SBuffer N 6.28</td>
<td>Buffer 6.281 s</td>
</tr>
<tr>
<td>Hard(P)</td>
<td>S</td>
<td>M</td>
<td>1.281</td>
<td>359.128</td>
<td>Buffer 1.281 s</td>
</tr>
<tr>
<td>Hard(P)</td>
<td>M</td>
<td>S</td>
<td>1.281</td>
<td>SBuffer N 1.28</td>
<td>Buffer 1.281 s</td>
</tr>
<tr>
<td>Hard(P)</td>
<td>M</td>
<td>S + RS</td>
<td>1.286</td>
<td>431.165</td>
<td>Buffer 1.286 s</td>
</tr>
<tr>
<td>Soft(I + P)</td>
<td>S</td>
<td>M</td>
<td>6.281</td>
<td>358.068</td>
<td>Buffer 6.281 s</td>
</tr>
<tr>
<td>Soft(I + P)</td>
<td>M</td>
<td>S</td>
<td>0.007</td>
<td>SBuffe N 6.28</td>
<td>430.212</td>
</tr>
<tr>
<td>Soft(P)</td>
<td>S</td>
<td>M</td>
<td>1.281</td>
<td>359.128</td>
<td>Buffer 1.281 s</td>
</tr>
<tr>
<td>Soft(P)</td>
<td>M</td>
<td>S</td>
<td>0.007</td>
<td>SBuffe N 1.28</td>
<td>431.165</td>
</tr>
<tr>
<td>Soft(P)</td>
<td>M</td>
<td>S + RS</td>
<td>1.286</td>
<td>431.165</td>
<td>Buffer 1.286 s</td>
</tr>
</tbody>
</table>

I = Inquiry, P = Page, M = Master, S = Slave, RS = Role Switch
N = # Handovers, SBuffe = Buffer for AP’s sensors

If the AP is a master then it should suspend ACL communications (typically by parking or holding current links) during 6.28 s for each time a mobile sensor performs handover and 1.28 s if inquiry is avoided. If master/slave role switch is performed the throughput of the AP is given by (4.12) and the required buffering time for the mobile sensor is 1.286 s after adding the eight slots [50] needed to perform role switch.

\[
Throughput_{RS}(AP_p) = \frac{(S_2.56 - S_p)(339B 86)}{10 S} = 431.165 \text{ Kbps.} \quad (4.12)
\]

- Bluetooth Soft Handover
  If an AP is a slave it is then involved in a scatternet while performing inquiry and page scans respectively. The throughput of the AP coincides with the throughput calculated for hard handovers given by (4.10). If the BDA is known, inquiry is not performed and the throughput of the slave AP is given by (4.11). As discussed above, buffering during inquiry and page is required for 6.281 s and 1.281 s depending if both procedures are needed or only page respectively.

If the mobile sensor is a slave then it will periodically execute inquiry and page scans obtaining a throughput given by (4.13) while connected to one AP, and obtaining a throughput given by (4.14) while connected simultaneously.
to two APs. The main advantage of this type of handover is the low latency (7.5 ms) achieved, which is given by (4.9).

\[
Throughput_{i+p} = \left(\frac{S_{2.56s} - S_p}{10s}\right) = 430.212 \text{ Kbps.} \quad (4.13)
\]

\[
Throughput_{i+p} = \left(\frac{S_{2.56s} - S_p}{12s}\right) = 179.034 \text{ Kbps.} \quad (4.14)
\]

If inquiry is not required the slave mobile sensor’s throughput is given by (4.15) while connected to one AP. While connected simultaneously to two APs the throughput is given by (4.16). Once more the main advantage of this type of handover is the low latency achieved given by (4.9).

\[
Throughput_{p} = \left(\frac{S_{2.56s} - S_p}{10s}\right) = 431.165 \text{ Kbps.} \quad (4.15)
\]

\[
Throughput_{p} = \left(\frac{S_{2.56s} - S_p}{12s}\right) = 179.564 \text{ Kbps.} \quad (4.16)
\]

If master/slave role switch is performed then (4.12) can be also used to determine the throughput of the AP. The required buffering time by the sensor including 8 slots needed to perform role switch is then 1.286 s.

### 4.8 Measurements

Measurements of the parameters described in Table 4.2 were performed in order to test practical implementations of Bluetooth handovers. Two types of measurements were conducted. On the one hand functionality measurements targeting system functionality constitute the first type of measurements. On the other hand parameter measurements targeting accurate measurements of throughput and handover latencies are the second type of measurements.

#### 4.8.1 Measurements Setup

Parameter tests were conducted using BC2 modules with CSR HCI v1.1 16.4 firmware. Functionality tests were conducted using connections at RFCOMM level using Linux Bluetooth stack. The system for functionality measurements was composed of the following parts:

- **MSC**
  
  The MSC was implemented with a computer that hold a database including all APs, BDAs of both APs and sensors, information about neighbor APs is also stored. The main task of the software running in the MSC is to decide which is the new AP in a handover and to control when the handover should be initiated.
• Access Points
APs were implemented using PCs running Linux with Bluez Bluetooth stack. APs are controlled by the server application, on start up the AP checks its neighborhood looking for other APs, the result is then sent back to the server application. They can periodically monitor RSSI or link quality or transmitted power to determine when a mobile sensor is getting out of range. They send a warning to the server application to notify about sensors moving out of range. The Bluetooth physical layer was implemented using Bluetooth v1.1 USB dongles from D-link.

• Backbone Network
The backbone network connecting APs and the MSC is an ethernet bus. The ethernet network was formed by a private hub in order to avoid undesired ethernet traffic from public networks, thus reducing collisions between ethernet frames.

• Mobile Sensors
Mobile Bluetooth sensors were emulated using a PC with a USB dongle. Page-scan mode was configured to mode R2 in all devices. Periodic inquiry and page scans were performed by the required devices, thus required devices were always discoverable independently of the Link Manager state they were placed. DH5 packet type was used for data transmissions. Devices were within line of sight and piconet radius was fixed to ten meters. No other Bluetooth device was active in the experiment area. Security aspects were kept out of the scope of the experiments, only BDA filtering was implemented at the MSC.

For functionality tests, handover algorithms have been implemented at application level following the design illustrated in Fig. 4.12.

For parameter measurements CSR BC2 modules implemented Bluetooth stack layers up to HCI. A PC was connected through the serial port to the Bluetooth module. The PC implemented the handover application. The Bluetooth stack distribution for parameter measurements is illustrated in Fig. 4.13.

4.8.2 Measurements Procedures
Two different measurement procedures were performed.

• Functionality Measurements Procedures
The network deployment illustrated in Fig. 4.14 was used under functionality experiments. The backbone network, APs and the MSC are static and hardwired via an ethernet network. The mobile sensor was placed on a mobile trolley and it was battery powered. The mobile sensor was displaced through a corridor at a speed of \( \approx 1 \) m/s. Moreover the mobile sensor was displaced only in one direction from one piconet until the handover was executed. Only two APs were deployed, thus AP election algorithm was not
tested and latency introduced by page fails was not considered, i.e; the new AP was always capable of connecting to the mobile sensor. RSSI is used as a handover decision key with a threshold level \( T \) empirically adjusted for the experiment deployment. Each AP issues the HCI \texttt{READ RSSI} command periodically with a 200 ms period to find out the strength of the received signal. When RSSI reaches the threshold value, the AP notifies it to the MSC.

- **Parameter Measurements Procedures**
  Tests were only focused on the throughput and latencies related to the mobile sensor network, thus QoS offered by APs was not tested.

  These measurements were performed by creating an ACL link at L2CAP level between a master and a slave. The slave was configured to perform periodic scans with a scan window of 10.625 ms. The slave transmits data at a constant rate of 92 Kbps using flow control. The master in its turn averages the number of received packets every second and calculates the throughput. Throughput during handover is measured with two masters connected to the same computer by two serial ports.

  Inquiry and page measurements were performed using a master connected to a computer. The computer sends HCI commands to the master to start inquiry and page procedures. Time is measured at the computer calculating the time difference between the time when the command is issued and the time when the respective completion event is received.
4.8. Measurements

Figure 4.13: Mobile sensor’s Bluetooth stack for parameter measurements.

Soft handover latencies are performed with two masters connected to the same computer, thereby sharing time base. The difference between the last packet received in the first AP and the first data packet received in the new AP is the handover latency.

Measurements were performed 100 times.

4.8.3 Functional Measurements

Functional measurements were carried out with Linux OS. Obtained results were not interesting from a parameter point of view since OS effects could not be avoided and resulted in a wrong order of packet delivery. But the functionality of the system was tested according to the following parameters [73].

- **Packet Losses**
  Packet losses were not recorded under any of the experiments.

- **MSC Algorithms**
  Handover initiations and executions conducted by the MSC were correct. Both soft and hard handovers with sensor as master and slave were tested with satisfactory results from a functional point of view.

- **AP Measurements**
  APs were in charge of detecting when sensors were moving out of range. All
algorithms described in previous chapters were tested obtaining satisfactory functionality.

4.8.4 Hard Handover Measurements

The parameter tests were focused on measuring the required time to carry out inquiry and page procedures involved in hard handovers. The inquiry was performed by issuing the HCI Inquiry command with a duration of 10.24 s and maximum number of responses equal to 1. The inquiry will terminate when the maximum number of responses is reached or when the time out elapses if no answer was received. Results of inquiry times are illustrated in Fig. 4.15. Paging times results when clock offset information is not available are illustrated in Fig. 4.16. Paging times results with clock offset information are illustrated in Fig. 4.17.
4.8. MEASUREMENTS

Figure 4.15: Inquiry time measurements.

Figure 4.16: Page time measurements without clock offset information.
4.8.5 Soft Handover Measurements

Parameter measurements are interesting when the mobile sensor is configured as a slave. Latency measurements results when the mobile sensor is configured as a slave are illustrated in Fig. 4.18. Throughput measurements results while handover is taking place and the mobile slave sensor is performing inquiry and page scans are illustrated in Fig. 4.19.

4.8.6 Results Analysis

The analysis of the obtained results is divided in the following sections:

- Inquiry and Page Times
  Inquiry results (max. 6.9 s) and page results (max. 2 s) fit the theoretical calculations presented in Table 4.1 and Table 4.2. It is remarkable that the use of clock offset information reduces the paging procedure as illustrated in Fig. 4.17.

- Sensor Throughput
  Sensor throughput for hard handovers have not been included in the results since disruptions in data transmissions occur while performing inquiry and page procedures. However for soft handovers the obtained results (max. 72.4 Kbps) differ from the theory (max. ≈ 179 Kbps). This is due to the fact that CSR HCI bt1.1 16.14 firmware changes automatically the packet
type from DH5 to DH1 when a slave is connected simultaneously to two masters. Thereby the result fits the theoretical value for DH1/2 throughput (86.4 Kbps) assuming a reduction due to scatternet scheduling, system latencies and scanning processes.

Moreover, the time that the slave sensor is involved in the scatternet is longer in practice (max. 264 ms) than in theory (≈ 7 ms). Thereby, during 264 ms the mobile sensor is transmitting data to both APs. The handover ends when the mobile sensor only sends data to the second AP. At this time the throughput decreases down to 72.4 Kbps. When the connection with the first access point is finished, the throughput is reestablished at its initial value (≈ 92 Kbps).

This is due to the fact that theoretical connection times do not take into account the internal Bluetooth module scheduling and the exchange of optional link configuration messages. Observed link establishment messages included connection request, connection accept, connection confirmation, read of supported features and change of maximum allowed slots among others.

- **Soft Handover Latency**
  The differences between the theoretical values of Table 4.2 (≈ 7 ms) and the obtained results illustrated in Fig. 4.18 (max. 264 ms) are due to the implemented scheduling in the Bluetooth module. Actually, the practical experiences do not show any correlation with the theory since data packets

Figure 4.18: Soft handover latency for a slave.
are interlaced with link establishment packets. Moreover, sensor’s data are addressed to both APs during handover, as described in the previous section.

4.9 Chapter Summary

In this chapter handover related effects such as throughput reduction and latencies have been investigated and tested. Investigations include calculations for performance evaluation. Several algorithms have been proposed covering soft and hard handover types.

For demonstration purposes the investigated algorithms have been implemented with a commercial Bluetooth platform from CSR. The validity of calculations was successfully confirmed against measurements. The conducted investigations and experiments lead to the conclusion that horizontal handovers (non inter-technologies) are possible with Bluetooth.

Obtained results show the importance and aid of the handover strategy election for optimization purpose of handovers. As described in Table 4.2, the selection of the handover strategy and type is related to trade-offs between QoS, sensor memory requirements and system design. If system QoS should be optimized then master/slave role switching must be supported and APs should be configured as masters. If sensor memory capacity is the parameter that sets design rules, then inquiry should not be performed. If constant data rate from a specific sensor is
desired then soft handovers with mobile sensor as slave should be implemented. If the mobile sensors should minimize their activity during handovers then NCHO strategies should be used with hard handovers.

The performed work has focused on the investigations for Bluetooth handover implementation however the following topics should be considered in real deployments:

- **Security**
  The performed experiments define a basic security level defined by BDA filtering performed by the MSC. Real implementations should add other security mechanisms such as encryption and pairing.

- **Deciding AP to Connect**
  The described algorithms assume that MSC has local positioning information for the election of the best fitting AP. The mobile sensor should be able to decide which AP is closer (higher RSSI) if master/slave role switching is allowed. Actual versions of the Bluetooth specification support Extended Inquiry Responses (EIR) including RSSI. Thus the mobile sensor could perform an inquiry obtaining RSSI values for all surrounding APs.

- **Device Inter-operability**
  If inter-operability is required, then MAHO and MCHO handover strategies should be avoided since they require extra functionality in the mobile sensors that is not defined by the Bluetooth specification. Moreover soft handovers with mobile sensors as slaves are based on slaves that can accept two Bluetooth links simultaneously, which is currently not supported by any profile.

- **Packet Losses**
  Mobile sensors should implement buffers for sensor data. If a master AP is performing inquiry or paging procedures, it will not address any sensor in its piconet, during this time, sensors must buffer data. The buffer size depends on how long time inquiry, paging and handover require. For example, if a master AP pages a new sensor performing handover it will not provide data transmissions to its slaves during 1.28 s, in the worst case a single mobile sensor belonging to the piconet will wait six consecutive inquiry and page procedures. The sensor should then buffer data for more than half a minute.

- **Seamless Handover**
  Seamless handover have been investigated [74] and successfully implemented at BB level in other works [75]. With this handover a mobile Bluetooth device believes that it is always connected to the same AP while changing APs. This algorithm copies the whole BB status of the current AP to a new AP creating a so called ”virtual connection”. This algorithm is an optimal solution for mobile sensor nodes but it is out of the scope of this thesis since the required access to the BB is not granted in commercial platforms.
• Limited Resources
  Bluetooth piconets are restricted to seven active slaves. If an AP has already
seven active links it will not accept a sensor being handed from a previous
AP. This case has not been covered by the investigations of this thesis.

• Ad-hoc Networking Handover
  The routing problem originated when sensors change their logical and physical
positions has not been covered in this chapter. The implemented work is based
on proxy servers implemented at APs. These proxy servers translate BDA
into IP addresses.

A distributed semi-static handover solution without a wired backbone network
have also been implemented [52] but it was not included in this chapter since
the routing problem is covered in Chapter 5. That distributed full wireless
solution does not need a LAN infrastructure and Bluetooth servers (MSC) to
perform hard handovers.
Chapter 5

Ad-hoc Routing with Bluetooth

Mobility of wireless sensor nodes in the network adds a significant challenge. After performing handover (as described in Chapter 4) data streams should continue being functional for both up and down links. A change in the physical position of a mobile wireless sensor will be accompanied by a change of its logical position in terms of network topology. Thus the logical path for reaching that node changes and a new route path is needed to successfully reach the sensor node. If the deployed network is an ad-hoc network, then ad-hoc routing algorithms are required. These algorithms will in some cases make use of metrics based on link quality or location as described in Chapter 3 to determine the best path between a source and a destination.

The study over MANETs is an entire field itself primarily focused for military utility [76], [77]. Results of works performed in this field include the development of well-known algorithms such as Dynamic Source Routing (DSR) and AODV. These algorithms are designed to cope with changing topologies [78], [49]. An exhaustive investigation of communications between any end-to-end node, in the case where all sensor nodes are mobile is out of the scope of the performed work in this thesis [52]. However the performed investigations and tests are applicable in WSNs which are predominantly static with few mobile sensor nodes.

This chapter investigates and describes the use of ad-hoc routing protocols in Bluetooth. The chapter starts by defining the problem of implementing ad-hoc routing protocols in Bluetooth. Secondly a review of existing ad-hoc routing protocols from a generic and a Bluetooth perspective is presented highlighting their performance characteristics. An overview of Bluetooth parameters that may add the design and implementation of ad-hoc routing algorithms follows. The main and novel scientific contribution of this chapter is the implementation of an ad-
hoc routing protocol layer and routing algorithm for Bluetooth which combines the DSR ad-hoc routing protocol with an on-demand Bluetooth scatternet formation protocol. The chapter ends with performance tests of the proposed algorithm and protocol.

5.1 Problem Definition

In mobile WSNs, nodes will change their position. If handover is performed based on infrastructure networks, the main challenge is how to route data packets from/to the sensor to/from APs. Thus the problem is reduced to find the right AP to reach a sensor.

However, in ad-hoc networks the lack of previous network infrastructure raises more difficult challenges. If a sensor node is to be communicated from any member of the network, then multi-hop route paths must be efficiently updated to maintain communications from/to the mobile sensor. Since Bluetooth does not define multi-hop routing, the designer faces the problem of how to use existing Bluetooth functionality to support such feature. Several routing protocols have been proposed for MANETs, but there is no straight solution applying them to Bluetooth due to Bluetooth’s specific BB and MAC-layer features. Moreover, in Bluetooth the problem is related to scatternet formation protocols combined with routing protocols. One valid approach for mainly static WSNs, is to first create a scatternet and then generate the routes. On the other hand if no scatternet is previously created finding route paths must be accompanied of scatternet formation. The last approach is suitable for mobile ad-hoc wireless sensor networks.

5.2 Goal: Provide Ad-hoc Routing Capabilities to Bluetooth

The goal in this thesis is to investigate and implement an ad-hoc routing protocol for Bluetooth to overcome the above described problems. The implemented protocol should be able to find any sensor in the WSN without the existence of a previous infrastructure. The protocol should work if no Bluetooth scatternet has been previously formed. Moreover previous works in the field base their solutions on simulations and theoretical Bluetooth behavior without providing routing effects in QoS [79], [80], [81], [82]. Therefore another important goal of the developed work here is to provide route formation time and supported QoS results by means of performance tests.
5.3 Existing Ad-hoc Routing Protocols

5.3.1 What is routing?
Routing is the technique that allows sending information from a source to a destination through a network. Typically, at least one intermediate node is encountered. Routing involves determining optimal routing paths and transporting information (usually referred as packets) through a network [83]. Differences between routing protocols include how to cope with asymmetric links, memory requirements, strategies for maintaining route information and discover methods. An overview of different existing routing protocols for MANETs follows focused on the two major classes which are table-driven and on-demand.

5.3.2 Table-Driven Routing Protocols
This class of protocols focuses on maintaining consistent and up to date information for each node to every other node in the network. Information is stored in one or more tables to save routing information. Changes in the network topology are reflected in routing tables by propagating updates. Updates are periodically propagated even if no event has occurred in the network to ensure network integrity. Thus routes to any node in the network should be valid at any given time [84].

In this sense, table-driven protocols are appropriate for large networks with evenly spread and constant traffic, where routes change constantly requiring low route discovery latency.

The main drawback of these protocols is the network overload generated by the propagating updates, which is consuming both bandwidth and energy. Moreover, the need of local sensor memory is increased by the fact that every single node must have information about the whole network topology.

From the scope of MANETs the permanent traffic generated of table-driven protocols together with related battery power consumption and memory requirements makes the election of these protocols an unwise solution.

5.3.3 On-Demand Routing Protocols
On-demand routing protocols only create routes when a source node needs to send data to a specific destination. This is why they are also called "source initiated".

The main drawback of this type of protocols is that routing information is not updated continuously and route discovery latency is higher than in table-driven protocols. Every node has its own routing table, which is first checked before sending a packet. If the route is not found in the table, it will then initiate a route discovery process in the network.

The main advantage of on-demand protocols is that changes in the network topology do not overload the network until a new route is needed [84]. Further-
more sensor memory requirements are reduced since sensors only save relevant routing information instead of a complete description of the network topology. These protocols fit in relatively small and not uniformly used networks.

From a MANET perspective, the reduced battery power consumption and the need of less memory resources make on-demand protocols an appropriate solution.

5.4 Existing Ad-hoc Routing Protocols for Bluetooth

Ad-hoc routing protocols for multi-hop WSN networks with Bluetooth involve route discovery and scatternet formation. There are two design approaches depending on the supported mobility by the network. On the one hand if the ad-hoc WSN is static, then a scatternet formation protocol is firstly performed and the routing protocol is secondly executed to update routing tables. The routing protocol is executed only once if no sensor nodes are added afterwards. On the other hand if the ad-hoc WSN is mobile, the routing protocol is firstly executed followed by the creation of a scatternet, an alternative is to combine both scatternet formation and routing simultaneously. The following protocols for Bluetooth cover some of the most representative designs for static and mobile ad-hoc WSNs.

5.4.1 On-demand Scatternet Formation

The on-demand scatternet formation protocol targets mobile ad-hoc WSNs based on Bluetooth [82]. This protocol proposes to combine scatternet formation with on-demand routing, thus eliminating unnecessary link and route maintenances. The created scatternet only involves nodes in the traffic route. Routes are created dependent on traffic demands, i.e., a route will only be created when a source node needs to exchange data with a remote destination node. This so called "scatternet route" interconnects adjacent piconets, thereby some intermediate nodes will be piconet bridges.

This protocol is based on some features not present in the Bluetooth specification such as Extended ID (EID) connectionless broadcast scheme. Performance results of this work are only based on simulations.

5.4.2 Dynamic Source Routing (DSR)

This protocol is suited for mobile ad-hoc WSNs. The DSR protocol [78] is an on-demand source routing protocol with no need of periodic broadcast information. It is also able to quickly adapt to changes in network topology with less memory requirements than table-driven protocols. This protocol is suited for reducing network overhead and conserve battery power. In the basic DSR protocol each mobile sensor keeps a route cache storing routes in the network. Routes are learnt by
receiving route discovery packets or by overhearing routing information from other packets. If a sender has a complete route in its route cache, the route is then used to transmit the data packet. If the route is not present, then a route discovery process is initiated. Once a route is found the route path is sent inside each packet’s header (explicit source routing). This is done to avoid loops, supporting multiple routes to a single destination. Intermediate nodes forward packets to the next hop. Route maintenance is performed to detect errors.

Route discovery is performed by broadcasting a route discovery packet (RDP) identifying the target (destination) sensor. A RDP consists of a destination address, a route record, the sources address, and a unique request ID. Each sensor node keeps a list of recently received RDPs and uses the unique request id and source address to detect duplicates. Intermediate nodes append their addresses and forward the RDP. When the destination receives the RDP it returns a copy of the route to the source.

The advantages of this protocol when adapted to Bluetooth include, no need of cost metrics, reduced route discovery time since nodes save the whole route record and not just one hop.

The main drawback is the packet overhead since routing headers include the whole route path.

This protocol has been adapted to Bluetooth in a previous work [85]. This work implements DSR discovery using an underlying already formed scatternet. The obtained results of this work are based on simulations.
5.4.3 BlueStars

This is a distributed scatternet formation protocol suited for static ad-hoc WSNs [86]. Bluestars is a multi-hop scatternet formation protocol which generates a mesh-like connected scatternet with multiple routes between pairs of nodes [87] as illustrated in Fig. 5.1. Assuming that not all nodes are within communication range of each other, all nodes participate in the formation of the scatternet. Nodes are entitled to keep track of their one hop neighbors.

5.4.4 BlueTree

Bluetree is a distributed and decentralized scatternet formation protocol for static ad-hoc WSNs that connects Bluetooth nodes in a tree like structure as illustrated in Fig. 5.2. Bluetree implements self-routing to handle new devices and the lost of old ones [88]. Each member of the tree must have a routing table. All traffic is routed towards the root of the tree. Thereby the nodes in the upper layers of the tree will consume their batteries faster. A disadvantage of the protocol is that if one parent is lost then all his children will not have a way to route their information and a tree rebuilding is needed.

5.4.5 BlueRing

Bluering is another scatternet formation protocol for static ad-hoc WSNs. The resulting scatternet topology is a ring as illustrated in Fig. 5.3. Each piconet has two slaves working as bridges to connect the downstream piconet with the upstream piconet respectively. The routing algorithm consists of sending packets around the
Figure 5.3: BlueRing topology.

Routing Support in Bluetooth

Currently, the Bluetooth specification does not define how to implement multi-hop routing in scatternets [50]. However the PAN profile v1.0 defines support for multi-hop routing in piconets. With the use of the PAN profile it is possible to create an ad-hoc piconet composed of devices that are capable of forwarding ethernet packets to any of the PAN users as needed. Exchange of data is carried out using the BNEP. However, these ad-hoc networks can not provide access to any additional network.

Bluetooth Ad-hoc Routing Protocol

The developed work comprises an on-demand scatternet formation protocol [89] and a routing protocol. Thereby the ad-hoc Bluetooth routing protocol combines parts of the basic DSR protocol [78] and the on-demand scatternet formation pro-
The main reason for implementing on-demand scatternet links is that if mobility is high in the network, the required time to create the scatternet is longer than the time needed to establish a single on-demand scatternet path. In this way the design differs from works where the scatternet is firstly created. The formation protocol only affects nodes in the multi-hop route between the source and the destination. Thereby a new scatternet will be created every time a data exchange session between two nodes of the network is started. In this sense both end-to-end scatternet and route creation are combined together. The protocol aims to reduce network overhead, power consumption and sensor memory requirements, therefore the chosen routing protocol class is on-demand. Thereby periodic propagation of potentially large routing updates through the network are avoided. The protocol is based on the following assumptions:

- The WSN is composed of $X$ nodes with $X > 8$ and not all of them are within range of each other, thus multi-hop communications are needed.
- All nodes in the network can be connected by means of a scatternet.
- All of the nodes in the network are relatively mobile and willing to forward packets for the others.
- The velocity of the nodes is moderate with respect to the packet transmission latency and range of the Bluetooth hardware.
- At the beginning, nodes do not have any information about other nodes participating in the WSN.
- Every node may communicate with any other node in the network.
- Nodes do not have a predefined Bluetooth role (master, slave or master/slave).

The basic form of the developed algorithm is presented in Fig. 5.4. When a source node is willing to transmit a data packet to a destination node it will first check the routing memory entries to find the route path. If the path is not found, then a flooding Route Discovery Packet (RDP) is sent. This algorithm is executed in a new layer placed between the application and the Bluetooth protocol stack. The routing layer has two inputs and two outputs. The inputs are frames from the APP layer and frames from the Bluetooth stack. The outputs are frames for the APP layer and frames for the Bluetooth stack as Fig. 5.5 illustrates. When a frame is received from the Bluetooth stack the algorithm implements the flowchart illustrated in Fig. 5.6. The on-demand route formation includes flooding for route discovery and backward scatternet formation. These steps are described in the following sections.
Figure 5.4: Bluetooth basic ad-hoc routing algorithm.

Figure 5.5: Bluetooth stack with routing layer.
Figure 5.6: Bluetooth ad-hoc routing algorithm (part 2).
5.6.1 Flooding-based Route Discovery

If no route information has been previously cached, then a RDP is flooded to the whole network to find the destination. The structure of a RDP is illustrated in Fig. 5.7. The field *Packet Type* is used to determine if the packet is data, a RRP or a RDP. The field *Route ID* identifies the route for the source. A RDP always includes two addresses at least, the *Destination Address* which corresponds to the 6-bytes BDA, and the source BDA address which is saved in the *Route Record* field. A new address will be appended in the route record field for every new hop the RDP travels. If a route involves \( X + 1 \) nodes then the size of the route record field is \( 6 \cdot X \) long.

Each node receiving the RDP will execute the algorithm illustrated in Fig. 5.6 [91]. In order to avoid loops, every node that receives a RDP containing already cached source and destination addresses, discards the RDP. This action not only avoids loops but lowers the network overhead due to flooding.

The problem is then how to generate flooding by broadcasting the RDP using Bluetooth. Since Bluetooth requires to establish a physical channel before transmitting data, a new connection must be done for every node before sending the RDP. The action described as “forward RDP” in Fig. 5.6 corresponds to the part of the algorithm illustrated in Fig. 5.8. Flooding will be performed every time a node does not have the necessary routing information to reach a destination.
Inquiry = N responses

Found destination ?

No

For X = 1; X < N+1

Yes

Connect to Node X

Append my BDA & Send RDP

Disconnet from node

X ++

No

Connect to Destination

Append my BDA & Send RDP

End forward RDP

Figure 5.8: RDP flooding algorithm.
5.6.2 Backward Scatternet Formation

When the first RDP arrives to the destination it is assumed that the RDP received corresponds to the shortest routing path. The destination generates then a Route Replay Packet (RRP) which is sent back along the newly discovered route [89]. The structure of a RRP is illustrated in Fig. 5.9. In this packet the Destination Address is equal to the Source Address or the RDP. The route path starts with the original destination address and includes all BDAs from intermediate nodes in downstream order. The RRP transmission is then accompanied by a scatternet formation as the flowchart of Fig. 5.11 illustrates.

In this way, every node in the routing path pages its downstream neighbor creating a physical link before sending the RRP. The physical link must be kept until the data session between the source and the destination is finished. When the data session is finished all Bluetooth links along the route are torn down. Moreover, during RRP forwarding every node in the path must save the route path in its cache. In this way only nodes participating in the routing path will save the routing information. Note that routing information is saved in the direction destination to source (downstream).

When the RRP reaches the source, all nodes in the path are interconnected as a scatternet with point-to-point links as depicted in Fig. 5.10. At this moment the first data packet from the source can be transmitted. The Bluetooth roles of every node in the route path are illustrated in Fig. 5.10.

<table>
<thead>
<tr>
<th>Packet Type</th>
<th>Packet Length</th>
<th>Reply ID</th>
<th>Dest. Address</th>
<th>Route Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>2 bytes</td>
<td>6 bytes</td>
<td>6 X bytes</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.9: A RRP packet.
Start send RRP to next hop

Connected to next hop? Yes → Send RRP to downstream hop
No

Page next downstream hop

Send RRP to downstream hop

Keep the Bluetooth Link

End send RRP to next hop

Figure 5.11: RRP transmission and backward scatternet formation.
5.6.3 Implementation

All sensors in the network must start enabling inquiry and page scans. Thus it is possible for other sensors to discover and connect to them. After that, the following actions may take place:

- Checking Cached Routing Information
  When a sensor node chooses to communicate with a remote node it firstly checks its cached route information. If the required information has already been cached, the source becomes the master and the first hop node becomes a slave to the source. The first hop node will then be a master to the next hop node and so on. The resulting scatternet topology is illustrated in Fig. 5.12. If the required information has not been cached, it will start flooding the network with the RDP to find the destination as Fig. 5.8 illustrates.

![Figure 5.12: Scatternet topology when route information has already been cached.](image)

- Flooding
  The source node will act as a master and will inquiry all its neighbors. When all answers from neighbors are obtained, it will search the destination among them. If the destination is found, it will establish a piconet with the destination and send the RDP thereafter. The destination will then answer back with the RRP. At this point data transmissions may start as depicted in Fig. 5.6. If the destination is not among the inquired nodes, then the master will construct the flooding packet including the destination address and its source address in the route record field. The master must page every single neighbor node to be able to transmit the RDP. Thereafter the master will disconnect from all neighbors and will wait for a RRP for the route associated to that destination. It is important to note that the master should buffer all data until the RRP packet its received.

Flooding will be continued by intermediate nodes repeating the RDP forwarding process. Every time a RDP is received, each intermediate node will append its own address to the route record field.

Furthermore, intermediate nodes will save the clock offset from devices paging. This is done in order to speed up paging procedures while creating the scatternet. Eventually the RDP will reach the destination node.

Intermediate nodes may check their cached routing entries when receiving a RDP, however in this work, this functionality has not been implemented.
• Backward Scatternet
  While flooding, when the last intermediate node finds the destination it will become the master of a piconet with the destination slave and it will forward the RDP so that the destination receives the address of the source. Thereafter, the destination will ignore further incoming RDPs with the same source address assuming that the first RDP received corresponds to the shortest path. The destination slave will then start the process of constructing the scatternet towards the source by sending the RRP. Intermediate nodes receiving the RRP will add the route to their cache and forward the RRP. An intermediate node receiving a RRP is automatically assigned the slave role for the previous hop node and the master role for the next hop node. The process will carry on until the last hop master will page the source which in its turn will be a slave as shown in Fig. 5.10. Note that links are not dropped after forwarding the RRP resulting in a linear scatternet.

• Data Transmission
  When the source node receives the RRP it can begin sending data to the destination. A header including the route path will be added to each transmitted data packet as shown in Fig. 5.13. All links must be kept until the source sends a data transmission end message. Once data transmissions are ended, all nodes participating in the route will keep the route information in their memory for future use. A drawback of using the packet format described by

<table>
<thead>
<tr>
<th>Packet type</th>
<th>Packet length</th>
<th>Route ID</th>
<th>Dest. Address</th>
<th>Route</th>
<th>Data payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>1 byte</td>
<td>2 bytes</td>
<td>6 bytes</td>
<td>6 X bytes</td>
<td>Y bytes</td>
</tr>
</tbody>
</table>

Figure 5.13: Data packet with routing header.

Fig. 5.13 is the header size which will reduce the sensor’s throughput. It is necessary to send the whole route so that intermediate nodes know which is the next sensor to route the information (source routing).

In the original DSR specification, the inclusion of the whole route in a data packet may also be used by all surrounding nodes which are in promiscuous mode. These nodes will "sniff" other routes communication packets so routes can be learned. The limitations imposed by Bluetooth in terms of broadcasting do not allow this functionality.

• Broken Routes
  If any node involved in the route moves away so that it can not be reached by its hop neighbors, the route will brake. In this case the last functioning hop will notify it to the source. Thereafter data transmissions are stopped and the source will restart a route discovery process. In this sense, the source must buffer transmitted data to achieve reliability. The implementation of
multi-hop acknowledgements and data buffering by the source is out of the scope of the developed work. However future works must include some kind of reliability procedures to ensure that no data is lost due to broken routes. Moreover route maintenance has not been implemented in this work.

- **Nodes Routing to Several Destinations**
  It is possible to support several routes simultaneously in a single device. If a sensor node is already active in a routing path, it can still perform inquiry and page scans and thereby receive new RDPs. However, when the node inquires and pages for flooding purposes, it will interrupt other active routes for a long time. This scenario has not been implemented in the current work, however, a mechanism to cope with this situation is described. A node participating in several routes will have to schedule data transmissions in the different active routes and when performing an inquiry or page, it should notify the previous and the next hop nodes about that situation. Thereafter the previous and next hop of the first route will interrupt data transmissions. In this way, the previous and next hop nodes will require buffers to store the information until inquiry and page are completed.

### 5.7 Resume of the Developed Ad-hoc Routing Protocol

According to the protocol described in previous sections the theoretical analysis of route discovery time is given by the following expressions.

- **Routing Header Size**
  In order to reduce route discovery and formation times it is assumed that RDP/RRP packets fit in a DH5 Bluetooth packet payload. Thereby 339 bytes minus four bytes for routing headers, can be used to include BDAs. Since BDAs are composed of six bytes, the maximum number of hops ($N$) is given by (5.1). One node must be taken away since the number of hops is equal to the number of nodes minus one.

  \[
  N = \left\lfloor \frac{335}{6} \right\rfloor - 1 = 54. \tag{5.1}
  \]

- **Worst Case Time for RDP Forwarding**
  Assuming inquiry results including seven slaves, the worst case for RDP forwarding occurs when every hop node must page seven slaves and the seventh is the one that will forward the packet that will arrive to the destination. The worst case time for RDP forwarding is given by (5.2) where $t_{RDP}$ is the time in seconds, $N$ is the number of hops, $I$ is the inquiry time, $n$ is the number of slaves, $P$ is the paging time, $RDP$ is the time needed to forward the RDP.
packet and \( d \) is the disconnection time. The result of 17 minutes is obtained assuming 10.24 s timeouts for inquiries, 1.28 s for paging, two slots for RDP transmission, seven slaves and two slots for disconnections.

\[
t_{RDP} = (N - 1) \cdot (I + n \cdot (P + RDP + d)) + I + P + RDP \approx 17 \text{ min.} \tag{5.2}
\]

- **Worst Case RRP Backward**
  The worst time case for RRP backward \( t_{RRP} \) is given by (5.3) where \( RRP \) is the time needed to forward the RRP packet.

\[
t_{RRP} = RRP + (N - 1) \cdot (P + RRP) \approx 1 \text{ min.} \tag{5.3}
\]

- **Total Route Discovery Time**
  The total route discovery time is the time between a source node starting to flood the network until the RRP is received by the source. This time is given by:

\[
t_{Total} = t_{RDP} + t_{RRP} \approx 18 \text{ min.} \tag{5.4}
\]

Once a route has been discovered and clock offset information is known a route reestablishment can be performed in:

\[
t_{reestablish} = (N) \cdot (P + RRP) \approx 37 \text{ s.} \tag{5.5}
\]

This result is calculated assuming paging times of 0.7 s with known clock offset.

### 5.8 Measurements

Measurements of the previously described protocol performance were achieved using a commercial platform in order to check the functionality and the characteristics of the protocol. The tests targeted route establishment time and effects in throughput.

#### 5.8.1 Measurements Setup

Performed measurements were based on CSR BC2 modules. Connections at L2CAP level were used for all tests. The designed routing layer was placed between the L2CAP and the APP layers as Fig. 5.14 illustrates. The measurement setup included:

- **Source Node**
  The source node was composed of a CSR BC2 module implementing the lower layers of the stack. The module was connected to a PC via RS232. The PC implemented part of the upper layers of the stack. The application, routing protocol and HCI driver layers were developed with NI Labwindows/CVI.
application layer was in charge of generating data packets that were passed to the route layer. The application offered a graphical user interface allowing the issue of several commands such as create route path, page, inquiry, and start data transmissions. The graphical user interface was also used to present results including inquiry time, page time, route creation time and throughput. The route layer implemented the protocol described in previous sections. The HCI driver supported a subset of HCI commands and events necessary to carry out the routing protocol.

- **Intermediate Nodes**
  Intermediate nodes were also based on CSR BC2 Bluetooth modules connected to a PC. The intermediate node was placed ten meters away from the source and the destination as Fig. 5.15 illustrates.

- **Destination Node**
  The destination node used a CSR BC2 module as well. The software in this node coincided with the one for the intermediate node. The destination node was placed ten meters away from the intermediate node and fourteen m away from the source as illustrated in Fig. 5.15. This was done to ensure that the source node could not find the destination when performing inquiry.

Figure 5.14: Bluetooth stack with routing layer.
5.8.2 Measurements Procedures

In order to test route creation times the nodes were static. The placement of the nodes corresponded to the deployment of Fig. 5.15. Inquiry times were measured performing inquiry from the source node. Inquiry-scans were configured with an inquiry-scan window of 10.625 ms and 2.56 s inquiry-scan interval. Page times were measured in a similar manner, page-scan mode was set to R2 in all modules (11.25 ms scan window and 2.56 s scan interval). Route path establishment times were measured by creating a route based on two hops corresponding to the intermediate and destination nodes respectively. Route throughput was measured in the destination node using DH5 packets. Measurements were performed 100 times.

5.8.3 Inquiry Time Measurements

In order to receive responses from most of surrounding nodes, the inquiry timeout is set to 10.24 s. Measurements performed show that the theoretical result fits the practice as shown in Fig. 5.16 with slight deviations. These deviations are due to system latencies introduced by the serial port driver and the OS.

5.8.4 Page Time Measurements

Page time measurements have already been presented in Fig. 4.16. The maximum obtained page time of 2 s is under the assumed theoretical value of 2.56 s.
5.8. MEASUREMENTS

5.8.5 Route Establishment Time

The total route establishment time results depicted in Fig. 5.17 vary between 22.1 s and 26.4 s. By reducing the inquiry timeout value to 5.12 s, the total route establishment time was reduced to 11.2 s as shown in Fig. 5.18. However, reducing inquiry timeout decreases the probability of finding all surrounding sensors in practical implementations. Inquiry never failed during performed experiments.

5.8.6 Throughput

Since there are only two hops, according to the protocol, the intermediate node is a master. Thus the resulting route is formed by a single piconet with one master connecting to two slaves supporting DH5 packets. In order to test the throughput in a scatternet the intermediate node (bridge) was configured as slave for the source and master for the destination. CSR bt1.1 16.14 firmware then allowed DH5 packets. The maximum obtained data rate was 14.5 Kbps.

5.8.7 Results Analysis

Obtained results are correct according to theoretical values with small deviations. In the case of inquiry deviations, the difference margin is due to latencies introduced by the serial port communication together with Bluetooth module and computer processing.
In the case of paging measurements, experimental results (2 s) are within theoretical values (2.56 s). The assumed theoretical value is based on the maximum value given by Bray in ref. [5].

Regarding route path formation times the obtained values (22.1 s - 26.4 s) are different to the theoretical calculations (24.3 s, assuming 1.28 s for page procedures), once more this is due to the variations of paging times depicted in Fig. 4.16.

Throughput measurements are highly dependent on packet type election. If for example asymmetric packet types are elected the route will present asymmetric behavior. However, the firmware adapts automatically the packet type depending on the quality of the link. Forcing packet types may result in even lower throughput. The obtained throughput (14.5 Kbps) is not inside the theoretical range (105.9 Kbps - 180 Kbps). The theoretical values are obtained for symmetric DH5 packet type, 240 bytes user payload and two guard slots. This big difference is attributed to the firmware’s scatternet scheduling algorithm and processing latencies.

It is remarkable that in practice, the route setup delay may be even longer than the obtained results because of the “disturbances” from the neighbors outside the shortest route. These neighbors may cause packet collisions during RDP and RRP delivery.
5.9 Chapter Summary

In this chapter the fundamental properties and features of routing protocols for mobile ad-hoc networks have been investigated. The results of the investigation lead to the design and implementation of an on-demand ad-hoc routing protocol for Bluetooth. This protocol is based on the DSR [78] protocol and the on-demand scatternet formation protocol [82] respectively. The main contribution of the performed work is then the combination of these two works providing a practical implementation. The obtained results complete previous work results based on simulations [82].

The protocol combines on-demand scatternet formation with route discovery (dynamic source routing, flooding-based route discovery, and backward scatternet formation) supporting multi-hop routes with traffic demands. Thus the route discovery process determines which sensor nodes will be part of the scatternet route. In this sense, there is no need of previous establishment of a Bluetooth scatternet prior to execution of the routing protocol.

The design of the protocol includes theoretical calculations of parameters to measure the performance of the protocol such as route establishment time. These calculations establish mobility limitations. The obtained limitations are related to the route establishment time in Bluetooth, which is relatively high due to inquiry and page processes.

Bluetooth needs to establish a link before exchanging of data can be carried out.
This implies that transmission of RDPs and RRPs can be achieved through a established link with the implicit delays that this operation adds.

Therefore the applicability of the developed protocol is restricted to low density networks requiring unbalanced and relatively low traffic with restricted mobility in terms of sensor velocity. For applications requiring high throughput with low delivery latencies it is more feasible to deploy static networks with mobility support restricted to a few nodes, e.g. high density scatternets with a scatternet formation protocol followed by a routing protocol, both executed once at network deployment time.

The routing process needs the scatternet structure information to complete the setting of routing tables in each relay sensor node. The combination of these two functions allows the use of joined protocol packets avoiding redundant control traffic.

An important characteristic of the protocol is that routes are discovered and used automatically fitting ad-hoc deployments. The protocol creates scatternets based on traffic demands, i.e. if data needs to be sent then a scatternet is created. Scatternets may be torn down when data sessions are finished, thus saving power if compared with static scatternets where links are kept active constantly. However the proposed protocol present several trade-offs:

- There is an obvious trade-off between saving power and reducing data traffic latencies. If routes are not torn down, power consumption will increase but data delivery latencies will be reduced.

- Another trade-off is established between the offered throughput and mobility support. If sensors frequently change their position then the network will be flled with RDPs reducing the throughput of already existing routes.

- Reducing inquiry timeout or maximum number of discovered devices will significantly reduce inquiry times and thereby route formation times. However the probability of not finding the destination node will increase.
Chapter 6

Bluetooth and Zigbee Mobility Comparison

Wireless Personal Area Networks (WPAN) technologies are the technology base for several WSNs systems. Both Bluetooth and Zigbee are WPAN technologies that have been applied to WSNs [8], [92], [93], [94]. The main reasons for adopting them in WSNs are their low power consumption together with their ad-hoc networking capabilities. The designer may ask himself, which of these two technologies is most suitable for WSNs. Since these two technologies are originally intended for different applications, it is not appropriate to compare them in absolute terms. However, comparisons studying the viability of their use for a restricted area can be performed. For example, in this thesis these technologies are compared from a mobility perspective. Thereby results of the investigations and experiments performed aim to compare the advantages and drawbacks of using these technologies in mobile WSNs. Related works in the area [95], [96] have not been focused in mobility and they lack experimental results.

This comparison is performed as a consequence of Bluetooth’s long network joining times, which affect both handover and ad-hoc routing performances. Since Zigbee allows data transmissions without TDM, it is a good candidate to reduce handover latencies and route discovery times. Moreover, the short network joining times together with embedded routing capabilities, and low power consumption offered by Zigbee makes its use in WSNs a wise solution.

This chapter starts with a short review of the main technological differences between Bluetooth and Zigbee. Investigations of Zigbee support for mobile applications are presented in the following section. The chapter continues by describing the designed algorithms for handover and positioning, as well as a discussion about Zigbee routing capabilities. A comparison of these algorithms with their Bluetooth counterparts is also provided. Finally, the chapter summary reports the main dif-
ferences between these two technologies from a mobility point of view in terms of performance.

6.1 Goal: Comparison from a Mobility Point of View

Several comparisons between these two technologies have been conducted discussing their strengths and weaknesses [95], [97], [98]. However, a comparison from a mobility point of view has not been yet carried out. Therefore, the main goal of this chapter is to provide a novel, fair comparison providing formal investigations and quantitative results. Obtained results and derived conclusions may be used as a design guide if mobility is a requirement in the WSN.

6.2 Main Differences within Bluetooth and Zigbee

Bluetooth and Zigbee are two similar technologies in terms of use of short range radio links and ad-hoc support. However, there are several differences between these two technologies. Some of the most prominent differences are listed below.

- From an Application Perspective
  Zigbee and Bluetooth are not competing technologies but solutions for two different application areas. Bluetooth is intended for personal devices such as cell phones and PDAs requiring transmission of files and larger data applications with a star master/slave topology. On the other hand Zigbee is intended for low duty cycle devices with short message applications supporting different network topologies [4]. Thereby Bluetooth is best suited for audio, personal devices, applications with larger data packets, non-critical battery life applications and non large networks. On the other hand Zigbee is more suitable for controls and sensors demanding low duty cycles, large density networks, with small data packets and long battery life applications [32].

- From a Battery Life Point of View
  - Duty Cycle
    Although Bluetooth supports three low power modes described in section 2.3.5, there is a trade-off between duty cycle and level of synchronization.
    Zigbee is designed to have a very low duty cycle including performing where both transmitter and receiver are in stand-by during 99 % of its operation. However, even in stand-by mode it consumes a certain power but it is much less than in active mode. The power consumption in stand-by mode is due to internal clocks maintaining, but the radio front-end is inactive.
6.2. MAIN DIFFERENCES WITHIN BLUETOOTH AND ZIGBEE

- **Active Mode**
  In Bluetooth, a slave can only transmit data after being addressed by the master, which requires all active slaves to listen for detecting packets addressed to them. Since the master’s scheduling layout for slave polling is not known by the slaves, active slaves must listen continuously.
  In Zigbee, the sensors (commonly end-devices) do not require polling from a coordinator or router, instead, end-devices may start a transmission of data whenever needed. Thereby they may switch off their front-ends when no data is transmitted. However the coordinator and the routers must keep their receivers on at all time.

- **Network Joining Process**
  Bluetooth inquiry and page times discussed in section 2.3.3 are longer than equivalent processes in Zigbee (as later discussed in section 6.3.2). Moreover synchronization requirements in Bluetooth make these processes consume more power.
  On the other hand, Zigbee network joining times are shorter and since tight synchronization is not required, the demanded power for performing such operations is lower.

- **From a Medium Access Perspective**
  Bluetooth uses TDMA for accessing the medium, i.e., every single Bluetooth device is granted a unique time frame which may be composed of one or several 625 µs long slots. During this time no other Bluetooth device participating in the same piconet is allowed to access the medium.
  Zigbee on the other hand uses carrier sensing techniques to access the medium. In particular CSMA-CA is implemented in Zigbee. CSMA-CA is a simple method for medium access that consists of listening before transmitting. If a device is willing to use the channel it will first listen to the channel to know if there is an ongoing communication in the same channel. If the channel is already in use then the device will wait for a random back-off time. If the channel is free it will get access to the medium.
  However Zigbee also supports TDMA, implemented by means of using GTS in beacon-enabled networks as described in section 2.4.2.

- **From a Spread Spectrum Technique Perspective**
  The modulation scheme in Bluetooth is GFSK. With this modulation data can be signaled at 1 Mbps. In order to spread the information in the frequency domain, frequency hopping is implemented, thus a Bluetooth front-end hops in frequency 1600 times per second.
  Spread spectrum is achieved in Zigbee by using DSSS wide-band technique. The DSSS process is applied before data modulation in the transmitter and after modulation in the receiver; it consists of another phase modulation at
a higher rate of the original BB data-rate. The result is a signal that has a wider spectrum with lower spectral power density.

- From a QoS Perspective
Bluetooth presents a deterministic behavior due to its TDMA technique. Slots can be granted to a certain device in a periodic fashion resulting in constant data rates in the absence of retransmissions. The higher achievable data rate with Bluetooth is 723.2 Kbps [5].

IEEE Std 802.15.4 does not provide isochronous communication, thereby communication is not periodically synchronized. However if low-latency communications are required, the PAN coordinator may grant Guaranteed Time Slots (GTSs) to avoid channel access delay. It is important to note that the use of GTS requires beacons which in their turn are only permitted in non mesh network topologies. The PHY layer provides a maximum of 250 Kbps for the 2.4 GHz band however, the data throughput might be significantly reduced depending on the established duty-cycle.

- From a Network Perspective
  - Network Components
    Two types of Bluetooth network devices are defined, the master which is responsible for starting links and establishing the synchronization in time and frequency, and the slaves.
    Zigbee presents three network components. One network coordinator which is needed to start a PAN by choosing the channel and the PAN-ID; routers which are FFDs for routing purposes and do not support low power modes, and end-devices which are RFDs supporting low power modes.
  - Star Topology
    Both Bluetooth and Zigbee support the star topology. In Bluetooth the center of the star is constituted by the master and devices around are slaves with a maximum number of active slaves equal to seven ($2^3 - 1$). In Zigbee the center device is represented by the coordinator with end-devices or routers around the coordinator. The maximum number of devices around the coordinator in a Zigbee star is $65534$ ($2^{16} - 2$).
  - Tree Topology
    This topology might be deployed with Bluetooth using scatternets. In Zigbee this topology is available when using parents and children. Several devices are connected to a parent device and each one of them can be connected to several children as depicted in Fig. 2.18. A device is only allowed to have one parent.
  - Mesh Topology
    This topology might also be implemented in Bluetooth as a scatternet.
However the Bluetooth specification does not define how this should be done. While in Zigbee this topology is defined and supported. With this topology, every Zigbee node (FFD) is able to communicate with any other node in its range.

6.3 Mobile WSN support in Zigbee

Mobility is not directly addressed in the Zigbee specification. However, Zigbee support for mobile WSNs can be defined as the offered support for location, handover and ad-hoc routing. Based on this definition the following sections investigate the applicability of Zigbee in mobile WSNs.

6.3.1 Local Positioning in Zigbee

Zigbee does not provide positioning services, neither a description of how they can be achieved with the use of existing functionality. However some parameters of the specification can be used for distance estimation as it happened with Bluetooth.

The main Zigbee parameters for distance estimation are the sensitivity, the output power and the mechanisms to read the received signal strength.

- Zigbee Sensitivity
  The standard defines a minimum sensitivity of $-85$ dBm in the 2.4 GHz Band.

- Zigbee Output Power
  The IEEE Std 802.15.4 requires that a compliant device must be able to transmit $-3$ dBm or 0.5 mW. Practical implementations offer higher powers (0 dBm). However, the specification recommends to minimize the output power to reduce interferences to other devices and systems.

- Receiver Energy Detection (ED)
  Receiver ED is intended to be used by the MAC layer as part of a channel election algorithm. ED values are an estimate of the received signal power within the bandwidth of an IEEE Std 802.15.4 channel. The value is represented by an integer with lowest value of 0x00 associated to less than 10 dB above the specified receiver sensitivity. ED values shall be linear and the range should be at least 40 dB with an accuracy of $\pm 6$ dB.

- Link Quality Indicator (LQI)
  The LQI measurement provides information about the strength and/or quality of a received packet. It may be implemented using receiver ED, a signal-to-noise ratio estimation, or a combination of these methods. The LQI measurement is performed for each received packet. The values of LQI are reported as an integer ranging from 0x00 to 0xff. Value 0x00 is associated with the
lowest LQ while value 0xff is associated to the highest. LQ values in between should be uniformly distributed and at least eight LQ values should be used.

6.3.2 Handover in Zigbee

The implementation of handovers is not addressed by the Zigbee specification. The main question however is: whether the term handover can be applied to Zigbee or not. Handover is defined as the mechanism to hand a communication link between two BSs. A Zigbee device only needs to join the network to be able to communicate with any of the network members. Thereby communications may be started at any time without the need of a previous a established link (once the device has joined the network).

Therefore the term handover can be restricted to a change of coordinator, i.e. inter-PAN handover. A change of coordinator may be desirable if the current channel’s LQ is low. However the specification does not cover this scenario.

Nevertheless intra-PAN handover can be considered when a mobile sensor gets out of range of another PAN member while streaming information to this PAN member. The specification covers this scenario by solving the problem with a routing protocol. In other words, a change in the sensor’s location is solved with a change in the routing path. It is important to note that only FFDs can start a route discovery processes for mesh networks.

The main parameters involved in hard Zigbee handover (inter-PAN) will be those related to network joining. Two parameters are related to network joining:

- **Association**
  
  Association refers to the operations performed when a router or end-device first joins a Zigbee network. The potential network device sends an association request to the PAN coordinator. The coordinator processes the request and decides if the device should join the network or not. The result of the request is notified to the potential network device by the PAN coordinator. More formally the whole procedure for an end-device or router associating to a PAN coordinator is illustrated in Fig. 6.1 [4]. Before sending an association request the potential network devices need to get information about the PAN. PAN related information is requested by the potential devices by sending a beacon request (active channel scan). The PAN coordinator in its turn will answer back with a beacon (independently if the network is beacon-enabled or not) containing PAN information such as use of beacons, GTS and PAN-ID. At this time the router or end-device has information about the stack profile, addressing mode, beacon order, if the sending device is a coordinator or another router and the most important, if the coordinator accepts association requests. Then the router or end-device issues an association request. The coordinator has then the responsibility to accept or decline association with that device. If a device is not in range of the coordinator it can get associ-
6.3. MOBILE WSN SUPPORT IN ZIGBEE

Joining Device  PAN Coordinator

Beacon Request  Beacon

Association Request  Association Response

Figure 6.1: Zigbee association chart.

ated by a router, in this case the router sends a beacon providing information about the PAN and notifying that it is not the PAN coordinator.

- **Binding**
  After association a binding between the desired clusters must follow to be able to communicate data. Binding is the process of connecting outgoing clusters with incoming clusters and it can be performed in two ways. The first binding method is referred to as "direct binding". Direct binding occurs when the destination address is known and nodes perform the binding by themselves, in this case the PAN coordinator is not required. In fact, even if the coordinator disappears from the network the bound devices will continue communicating. The second binding method is the "indirect binding". This type of binding takes place at the PAN coordinator. In this case the PAN coordinator is required to perform communications between the two network devices.

### 6.3.3 Ad-hoc Routing in Zigbee

One of the strengths of Zigbee is the inclusion of an ad-hoc routing protocol. Ad-hoc routing is intended for mesh Zigbee networks, if mesh networks are not allowed, then tree routing is the default algorithm. The ad-hoc routing protocol is based on the following aspects.

- **Address Assignation**
  Each Zigbee device has a unique 64-bit IEEE MAC address. This address is used at network joining time to associate devices with a PAN coordinator or router. During association the PAN coordinator or router assigns a 16-bit network address to the device. This 16-bit network address will be further
used in network communications and routing processes. By default these short network addresses are assigned using a distributed addressing scheme. This scheme provides every potential parent (in a tree topology) with a finite sub-block of network addresses that can be assigned to future children as shown in Fig. 6.2.

- **Routing Cost**
  Alternative routes are compared using a path cost metric during route discovery and maintenance. This metric is calculated using the link cost of each one of the links in the path [32]. All links costs are summed for a route obtaining the cost for the path given by:

\[
C(P) = \sum_{i=1}^{L-1} C\{D_i, D_{i+1}\},
\]  

(6.1)

Where \(C(P)\) is the path cost, \(C\{D_i, D_{i+1}\}\) is a link cost, and \(L\) is the number of devices in the path. When a route is being discovered each new path cost received is compared with the actual cost of the route, if lower costs are received then routing tables are updated.

- **Routing Tables**
  FFDs may contain routing tables in order to store route information. A Zigbee routing table is illustrated in Table. 6.1.
6.4 Distance Estimation with Zigbee

The developed distance estimation algorithm is based on its Bluetooth counterpart described in section 3.7.

6.4.1 Zigbee Mobile Radio Propagation Model

The algorithm uses a Zigbee mobile radio propagation model which in turn is based on the Bluetooth mobile radio propagation model detailed in section 3.5. Since Zigbee operates in fourteen channels of the 2.4 GHz ISM band, the wavelength ($\lambda$) may be calculated at the center frequency of each separate channel. Thus for the first channel the center frequency is 2.405 GHz [4] obtaining $\lambda = 12.47$ cm.

Combining this value with (3.4) the simple expression of (6.2) is obtained.

$$PL = 40 + 10n \log_{10}(d).$$

(6.2)
6.4.2 Algorithm for Distance Estimation with Zigbee

Finding the distance in (6.2) the estimated distance is given by:

\[ d = 10^{\frac{PL-40}{10}} \text{ with } PL = P_t - P_r. \]  

(6.3)

It is important to remark that \( P_r \) is obtained from the LQI field received with each packet at MAC level or the ED value, then \( P_r \) is obtained by:

\[ P_r(dBm) = (LQI \text{ or } ED) - Offset. \]  

(6.4)
6.4.3 Measurements

Two Zigbee devices from TI-Chipcon (CC2420) are connected with one meter distance between them. One of the devices is configured as PAN coordinator and the other as end-device.

The end-device is mobile and is displaced after connection along a 80 m long corridor. Only one parameter is measured in this case, ED. This parameter is obtained by the RSSI (ED) information. In order to implement the test TI-Chipcon’s CC2420ZDK range demo application is run. The end-device sends packets to the coordinator; the coordinator in its turn sends packets back as a response. Thereby the end-device can obtain the LQI and ED values. The RSSI information is collected with the help of a Zigbee packet sniffer also provide with TI-Chipcon’s CC2420ZDK. The default output power is configured to -3 dBm in both devices. Measurements were performed 100 times in the same environment that measurements for Bluetooth were performed, that is, indoors with direct line of sight and antennas oriented for maximum reception level. Thereby the theoretical calculations for the Zigbee propagation model are obtained with a path-loss coefficient equal to 2.7. Illustrated results in Fig. 6.4 show worst case measurements.

6.4.4 Accuracy Study

The accuracy of the distance estimation algorithm is determined comparing the theoretical calculated values with Zigbee mobile radio propagation model, with the
worst case measurements.

- The accuracy for RSSI based measurements is 50 m (inside a range of 80 m) in the worst case. This low accuracy is due to supported accuracy of the Zigbee specification ($\pm 6$ dB), thereby 12 dB of uncertainty are obtained.

### 6.4.5 Comparison with Bluetooth

Zigbee measurements have less accuracy than obtained results with Bluetooth. The measurements range in Zigbee is greater. This is due to the fact that Zigbee does not have a golden range like Bluetooth does, therefore RSSI readings do not present flat regions like it happens with the GR in Bluetooth.

However obtained results are highly dependent on manufacturer’s specific implementation including RSSI readings accuracy, and antenna design. Therefore the general conclusion is that both technologies may be use to get rough distance estimations indoors but not to implement high precision local positioning systems.

### 6.5 Handover for Zigbee

Zigbee provides two parameters that may be used in conjunction with (4.1) to implement a handover algorithm, LQI and receiver ED. These parameters and their implications have been described in section 6.3.1. For example, LQI readings are
translated to RSSI values and then compared with the threshold level \((T)\) calculated in (4.1) to determine when the handover should be initiated.

### 6.5.1 Algorithm for Inter-PAN Handover

Inter-PAN handover is defined as the process that hands an associated network device from one PAN coordinator to a different PAN coordinator as the scenario of Fig. 6.5 illustrates. The IEEE Std 802.15.4 covers this scenario with orphan devices, a device becomes an orphan after determining that it has lost connection with its PAN coordinator [4]. Thereafter orphan device will start an orphan channel scan to locate beacons of available PANs. In order to reduce handover latencies, the orphan method will be avoided. Since the Zigbee specification does not support devices participating in two or more PANs simultaneously, hard handovers are required. Moreover the mobile sensor \((M)\) will initiate and execute the handover so the strategy is MCHO. The handover algorithm will first disassociate the mobile sensor from the first PAN coordinator and then will request for association to the new PAN coordinator as the flowchart of Fig. 6.6 describes. The handover latency of this algorithm is then given by (6.5) where \(L\) is the latency, \(t_D\) is the disassociation time, \(t_A\) is the association time and \(t_B\) is the binding time.

\[
L = t_D + t_A + t_B.\tag{6.5}
\]
This type of handover is applicable to both RFDs and FFDs.

6.5.2 Algorithm for Intra-PAN Handover

Intra-PAN handover is defined as the process that changes the first hop destination address of a mobile device inside a PAN as shown in Fig. 6.7.

This handover is required when a mobile sensor changes its location and gets out of range of its first-hop device. The Zigbee specification covers this scenario by its routing maintenance and repair algorithms. However, to reduce handover latencies, route request may be issued before route maintenance is needed. The algorithm for intra-PAN handovers is illustrated in Fig. 6.8.

6.5.3 Measurements

- Measurement Setup
  For inter-PAN handovers the parameters to be measured are association and binding. Association and binding are measured by starting a PAN coordinator followed by the start of an end-device which performs association and binding operations. Joining times are measured running TI-Chipcon’s CC2420ZDK kit range demo application by logging the association times from Z-Trace tool. The association time is then measured as the time difference between a reset confirmation and an association response for the end-device.
6.5. HANDOVER FOR ZIGBEE

Figure 6.8: Zigbee intra-PAN handover algorithm.

Figure 6.9: Zigbee inter-PAN handover measurement setup.

Binding times are measured with the same test bench used for the association time measurements.

Handover time is tested using two PANs as illustrated in Fig. 6.9 where an end-device transmits data to a coordinator until it gets out of range of the coordinator. Then the end-device performs a software reset and restarts its application associating with the coordinator of the new PAN.
Figure 6.10: Zigbee association times with three beacon requests.

Figure 6.11: Zigbee association times with one beacon request.
• Results
  
  Association Times
  By default TI-Chipcon CC2420 transceivers perform three consecutive beacon requests. This is done to calculate an average of the LQI for each beacon sending device, thereafter the device with highest LQI is chosen. The obtained results are illustrated in Fig. 6.10. The worst association time during the experiments was 1.28 s.

  In order to reduce handover latencies the number of beacon requests was reduced to one. Thereby association times can be reduced as shown in Fig. 6.11. The resulting association times are approximately the half of previous results depicted in Fig. 6.10.

  Binding
  Obtained binding times are illustrated in Fig. 6.12. The performed binding is direct (also called auto-match). In this case the device willing to bind sends a broadcast with its description and the targeted cluster and endpoint respectively. Thereby allowing direct communication without the coordinator. If the targeted device receives the auto-match request, it will then answer back resulting in a connection of the specified clusters.

  Inter-PAN Handover Time
  The inter-PAN handover times are slightly higher than the sum of associ-
ation times and binding times, this is due to the software reset performed by the mobile sensor. The handover times for associations with one beacon request and binding with auto-match are depicted in Fig. 6.13.

– Intra-PAN Handover Time
Intra-PAN handovers were tested in the scenario illustrated by Fig. 6.14. A mobile router firstly associates to another router, and starts data transmissions addressed to the coordinator. Thus data are routed via the static router (R1). The mobile router (R2) is then displaced towards the coordinator. When the LQI threshold is reached, then the mobile router sends a RREQ packet. When the coordinator answers with the RREP packet, data from the mobile router is sent directly to the coordinator. TI-Chipcon’s CC2420ZDK transmit demo application was used during the experiments. In this application the end-device sends a message containing 90 bytes as fast as possible. Handover times are measured with a sniffer provided with TI-Chipcon’s CC2420ZDK kit. The handover time is the difference between the time when the mobile sensor sends the last data packet before sending the RREQ packet and the time when a data packet is directly delivered to the coordinator. The longest obtained handover time for this scenario is 74 ms. If the handover is not implemented then, by default a Zigbee device will try to retransmit the desired message a number of times before issuing a route request command, measurements revealed that a Zigbee device with TI-
Chipcon’s Zigbee stack keeps retransmitting during 1.2 s before issuing the route request command.

### 6.5.4 Handover Latency Study

Taking into account the worst cases for associating and binding during the experiments, the total joining time can be up to 0.8 s when using one beacon request. However it should be noticed that present experiments were conducted with point-to-point connections and no other Zigbee device was involved in the communication. In other words, if several devices are willing to join the network at the same time, they will follow the CSMA/CA scheme [4] to get access to the channel, this may result in longer network joining times.

For intra-PAN handovers the implemented algorithm reduces the handover time down to 74 ms. This is due to the fact that the route request command is issued directly without waiting for a number of failed retransmissions (it avoids route maintenance).

### 6.5.5 Comparison with Bluetooth

The maximum Zigbee hard handover time obtained in this work (0.8 s with one beacon request) is lower than obtained results for MCHO hard handovers with Bluetooth (up to 8.9 s including inquiry) as described in section 4.8.6. The short network joining times that Zigbee provides are the main reason of its better behavior in hard handovers. However in high dense Zigbee networks this time may be increased due to the CSMA-CA algorithm.

Bluetooth soft handover latency (264 ms) is higher than intra-PAN Zigbee handover latency (74 ms). This is due to the fact that it is faster to change the route path than to establish a new link. However, this observation must take into account the experiment scenario, if longer routes are involved, then the soft handover time for Zigbee will increase.
CHAPTER 6. BLUETOOTH AND ZIGBEE MOBILITY COMPARISON

6.6 Performance of Zigbee Routing Protocol

Modifications and tuning of the original Zigbee routing protocol are out of the scope of this thesis. Instead, experiments for testing the protocol have been conducted. The results of these experiments are compared with Bluetooth results in terms of route formation time and throughput.

6.6.1 Measurements

The measurements conducted are targeted at the points of establishment time and route throughput.

- Measurements Setup
  The routing experiments were based on TI-Chipcon’s CC2420ZDK kit transmit demo application. The application was executed without implementing application level ACKs in order to provide a fair comparison with Bluetooth. The deployed network topology for routing experiments is shown in Fig. 6.15. The second router (R2) will follow the ad-hoc routing algorithm described in section 6.3.3 in order to find the path to reach the coordinator. Route formation time is measured with a Zigbee sniffer provided with TI-Chipcon’s CC2420ZDK kit. Thereby route formation time is the time difference between the starting of a route discovery process from the second router (R2) and the reception of a route reply packet from the first router (R1).

- Results
  Routing formation is started by the second router (R2) by issuing a route request, the process ends when the first router (R1) eventually responds with a route reply. The obtained route formation times are depicted in Fig. 6.16.
6.6. PERFORMANCE OF ZIGBEE ROUTING PROTOCOL

Figure 6.16: Zigbee routing formation time for two hops.

Figure 6.17: Zigbee throughput for a two hops route.
• Throughput
Since the information must be routed via the router R1, throughput is reduced. Throughput measurements with a route of two hops are illustrated in Fig. 6.17. The maximum achieved throughput was 51.4 Kbps.

6.6.2 Comparison with Bluetooth
Ad-hoc routing in Zigbee is supported by the standard. The network layer implements the routing protocol thereby routing issues are transparent for the designer.

The facts that Zigbee uses CSMA-CA and supports fast broadcasts, reduce route formation times significantly if compared with Bluetooth. In the example of a network with three nodes and two hops, routing formation takes 0.123 s in Zigbee while it takes 26.4 s for Bluetooth for a similar scenario (using the DSR adapted routing protocol of Chapter 5).

Regarding throughput in a route, the highest data throughput achieved for the above described scenario is 51.4 Kbps while in Bluetooth, throughput up to 14.5 Kbps can be achieved.

It is important to notice that with the present Zigbee specification only FFD devices can issue the route request command.

6.7 Chapter Summary
In this chapter Zigbee mobility related aspects such as positioning, handover and ad-hoc routing have been investigated and tested. Investigations include a review of parameters providing information about signal strength; calculations of handover latencies for intra-PAN and inter-PAN handovers and analysis of the Zigbee routing protocol.

The novel proposed handover algorithms and the comparison itself are the main scientific contributions achieved.

For demonstration purposes the investigated algorithms have been implemented in a commercial Zigbee platform from TI-Chipcon. The validity of the calculations was successfully confirmed against measurements.

The performed comparison is based on empirical data and can be summarized in Table 6.2, where for each Zigbee mobility aspect its Bluetooth counterpart is presented.

The conducted investigations and experiments lead to the following conclusions:

• Regarding routing capabilities the conclusion is that Zigbee is more suitable for mobile WSNs than Bluetooth if a mesh network is to be deployed, since logical data paths in the network are automatically achieved by the ad-hoc routing protocol. Moreover route formation times are dramatically reduced due to the fast broadcast nature of Zigbee.
### Table 6.2: Bluetooth and Zigbee mobility parameters summary.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Bluetooth</th>
<th>Zigbee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positioning support</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Non std. Distance accuracy</td>
<td>≈3 m (Range 10 m)</td>
<td>≈50 m (Range 80m)</td>
</tr>
<tr>
<td>Accuracy (standard)</td>
<td>2 dB ($T_x$ Power)</td>
<td>± 6 dB (ED)</td>
</tr>
<tr>
<td>Parameters</td>
<td>RSSI, Tx Power</td>
<td>LQI, ED</td>
</tr>
<tr>
<td>Handover support</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Network joining time (theory)</td>
<td>≈ 12.8 s - 33.28 s</td>
<td>≈ 0.3 s</td>
</tr>
<tr>
<td>Non std. hard HO latency</td>
<td>≈ 8.9 s</td>
<td>≈ 0.8 s</td>
</tr>
<tr>
<td>Non std. soft HO duration</td>
<td>≈ 0.264 s</td>
<td>≈ 0.074 s</td>
</tr>
<tr>
<td>Ad-hoc routing support</td>
<td>No(*)</td>
<td>Yes (FFDs)</td>
</tr>
<tr>
<td>Route formation time (2 hops)</td>
<td>≈ 26.4 s (Non std.)</td>
<td>≈ 0.123 s</td>
</tr>
<tr>
<td>Route throughput (2 hops)</td>
<td>≈ 14.5 Kbps</td>
<td>≈ 51.4 Kbps</td>
</tr>
</tbody>
</table>

* Available in PAN profile, applicable for only 1 piconet.

- In the case of hard handovers, the short Zigbee network joining time reduces the handover latency by eight times. For the developed soft handovers, Zigbee also presents lower handover latency. An important aspect related to the handovers is the power consumption. Observations during experiments showed that Zigbee required half of the source current than Bluetooth (≈ 50 mA) while joining a network. This factor should be taken into account if handover frequency is high.

In summary, Bluetooth is more suitable for applications demanding continuous large data transmissions, rechargeable batteries, with low density networks and star topologies. Mobility is constrained in ad-hoc networks by the long Bluetooth route formation times. However Bluetooth can be deployed in mobile WSNs where most of the nodes are static and a few are mobile or in infrastructure based networks.

On the other hand Zigbee is more suitable for applications with short messages, aperiodic data with low duty cycle, long battery life and large networks supporting mesh topologies. The short handover latencies together with the supported ad-hoc routing protocol makes Zigbee a better technology for mobile ad-hoc WSNs.

After all, both technologies support a certain level of mobility and it is up to the designer to decide which technology fits better in its application, keeping in mind trade-offs between, throughput, battery life, cost, handover rate and sensor resources.
Chapter 7

Concluding Remarks

The main focus of this thesis has been on analysis, design and implementation of algorithms and protocols enhancing the mobility support in short range radio standards. Aspects such as local positioning, handover and ad-hoc routing have been considered.

The need for mobility in WSNs makes it necessary to develop algorithms and protocols which combined with standard protocol stacks and front-ends can offer seamless connectivity. In this work, Bluetooth and Zigbee standards are targeted for implementation of the algorithms and protocols. The essential differences between the implications of these two technologies have been addressed. It can be concluded that standard functionality offered by Bluetooth and Zigbee is not sufficient to provide mobility while maintaining high QoS in the WSN. Therefore alternative mobility support techniques have been investigated, such as the implementation of local positioning.

Performance measurements of the designed local positioning algorithm for Bluetooth show that distance estimation techniques based on signal strength present limitations, since propagation characteristics are dynamic and uncertain. This is due to the drastic differences in the signal strength or quality of the communication link obtained for small changes in position or direction of the front-ends. However obtained results are applicable in handover, network topology construction and efficient routing path formation. Various distance estimation methods have been studied for Bluetooth. On the one hand the use of Bluetooth RSSI is limited by the specification and is very dependent on manufacturer implementation. On the other hand the use of transmitted power is supported by the standard but restricted to class 1 devices. The election of RSSI or transmitted power for class 2 and 3 devices presents a trade-off between range of measurements and accuracy in practical implementations. In summary, the obtained results open a range of applications where local positioning with low accuracy is required. For example, it is possible to estimate the trajectory of a mobile sensor. This trajectory information may be
CHAPTER 7. CONCLUDING REMARKS

processed by an automatic door for aiding disabled persons in building access. This trajectory information may also be used for predicting handovers and best route paths. The main contribution and difference with previous works is the description of a location system covering all Bluetooth device classes. This is done with the election of the distance estimation method (RSSI or transmitted power).

Handovers have also been investigated and tested with Bluetooth. The use of soft handovers is a promising solution for the reduction of handover latencies, thereby enabling continuous data transmissions which are desirable in applications such as telemetry. In order to aid the design and improve the handover performance in Bluetooth, theoretical studies for several handover types with their impact on QoS and sensor memory demands have been carried out in this work. These theoretical studies together with the experimental results constitute the main difference with previous works in the area. The proposed handover types cover a wide variety of WSN topologies based on Bluetooth. However, integrating these functionality with the standard is a challenging task. Therefore, in such cases, it might be more appropriate to use hard handovers at the cost of increasing handover latencies and thereby sensor cost due to required memory resources.

The last work developed for Bluetooth deals with multi-hop ad-hoc routing protocols. In order to design and implement such functionality, two different approaches may be considered; the scatternet formation approach with table-driven routing, and the on-demand routing approach. The table-driven approach is not sufficient for mobile WSNs, since its route maintenance tasks will reduce the overall QoS in the WSN plus it will also increase the power consumption. The on-demand approach on the other hand, is more suitable for mobile environments reducing overall route maintenance, thereby increasing battery life and QoS. The proposed on-demand routing protocol combines results of two previous works, the DSR protocol [78] and the on-demand scatternet formation [82] protocol. The combination of these two works results in the design (with theoretical performance analysis) and implementation of a new routing protocol for multi-hop Bluetooth-based WSNs. The most attractive feature of the adopted design is that the route path and the underneath scatternet are created simultaneously based on traffic demands thus reducing the network overhead. The principal outcomes of this part are the description of a new on-demand ad-hoc routing protocol together with the characterization through experimentation of routing effects in terms of route formation time and route throughput. The performed experiments complete previous work results based on simulations. Obtained results expose the long route formation times required in Bluetooth due to its broadcast limitations, thus constraining mobility in Bluetooth ad-hoc WSNs.

The last part of this work presents a comparison between Bluetooth and Zigbee in terms of mobility support. The performed investigations with Zigbee are novel as well as the designed and implemented handover algorithms. Although Zigbee does not provide a handover mechanism, nor local positioning services, its MAC capabilities and asynchronous communications situate this technology in a more advanced
position than Bluetooth for mobile ad-hoc applications. The implications of these capabilities are translated to shorter network joining times, and broadcasting easiness, thereby reducing both handover latencies and route formation times. Both intra and inter PAN handovers have been investigated and implemented providing reductions in handover latencies. Inter-PAN handover may be used to improve QoS, e.g., if a PAN suffers a lot of interference, the sensor node may decide to change to a coordinator in another channel where interference is lower.

From all performed analysis and experiments it can be concluded that Bluetooth can be applied in mobile WSNs with the aid of the developed algorithms and protocols. These algorithms and protocols provide the necessary mechanisms to support mobility while keeping QoS within acceptable levels. In the case of Zigbee, the specification already provides mechanisms to cope with mobility related issues (orphan procedures, route maintenance), however the developed algorithms (handover) increase the QoS. Thereby both Bluetooth and Zigbee are suitable for mobile WSNs with certain restrictions. These restrictions are related to the existing trade-off between grade of QoS in the WSN and the grade of supported mobility. As an example, Bluetooth-enabled sensors with high velocity will not be able to find ad-hoc routes to desired destinations due to the long required route formation time. On the contrary if the deployment is based on infrastructure networks a fast, soft handover will enable relatively high velocities of the sensors. Thereby Bluetooth will be more suitable for infrastructure networks supporting a moderate grade of mobility and also for ad-hoc networks with a reduced mobility grade. On the other hand Zigbee is more suitable for ad-hoc networks with a moderate grade of mobility and reduced throughput. It should be noticed that these interesting and remarkable observations are only possible through the MAC level analysis carried out.

Theoretical analysis presented in this work have been verified against measurements on commercial standard compliant platforms. The implemented algorithms covering local positioning, handover and ad-hoc routing, confirm the theory with very good agreement. As a demonstration of the importance and aid of the theoretical analysis for optimization purpose of QoS with mobility algorithms, a soft handover algorithm based on RSSI measurements has been implemented for the M3C project with superior handover performance in terms of handover latency and offered QoS (compared to standard hard handovers).

Finally and from a sensor architecture point of view, it should be notice that the implemented stand-alone Bluetooth sensor architecture based on CSR virtual machine applications, enables embedded wireless sensors with a reduction of discrete components. Thus providing a standard compliant solution reducing cost and power consumption.
7.1 Future Research & Improvements

Even though obtained results of this work offer improved performance for mobile WSNs based on Bluetooth and Zigbee, a number of interesting research topics are remaining in this field and in this section some of the ideas are revealed.

- **Local Positioning with Bluetooth**
  For local positioning with Bluetooth it is of great interest to implement the functionality using distributed algorithms in ad-hoc WSNs [99]. Investigations and design in this thesis are oriented to absolute positioning for infrastructure networks where the position of the landmarks is known ahead. In ad-hoc networks the algorithm should establish relative positions. These relative positions may be used by protocols such as LEACH [28] to deploy energy effective WSN topologies.

  The required time to obtain RSSI information from surrounding nodes can be reduced with the use of inquiry results with RSSI. This functionality is supported in newer versions of the Bluetooth specification. In order to guarantee inter-operability, the algorithm should be used in combination with the local positioning profile. The local positioning profile is only available as a draft version from the Bluetooth SIG. This profile supports time as a measuring method. Thereby investigations of how to implement distance estimation based on time measurements are needed.

- **Handover with Bluetooth**
  As described in Chapter 4, seamless handovers may be achieved if the algorithm is executed at BB level instead of application level. Thereby, research is required to study how to synchronize APs and exchange BB status.

  For hard-handovers the use of inquiry results with RSSI can be used as aid when choosing the next AP. This will improve the link quality for the next connection.

  In the case of soft handovers, scatternet scheduling will increase sensor throughput while handover is executed.

  Another topic, which has been ignored in Chapter 4 is handover in ad-hoc WSNs. In this case, both handover and route establishment should be performed simultaneously. Finding routes in advance will decrease handover latencies.

- **Ad-hoc Routing with Bluetooth**
  The main limitation imposed by Bluetooth for ad-hoc routing is the long route establishment time as described in Chapter 5. This time is mainly dependent of inquiry and page times. An alternative to reduce the route establishment time is to use the inquiry stage to transmit the RDP in a broadcasting fashion.
This could be done by using an extended ID (EID) connectionless broadcast scheme which achieves a very much shortened route discovery delay [82].

An improvement in QoS and immunity against interference may be achieved if slave-slave bridges are used. This will reduce the number of piconets in the scatternet, thereby reducing interference and scatternet synchronization requirements.

The implemented protocol in Chapter 5 does not support asymmetric routes. In a mobile wireless ad-hoc network uni-directional links may be limited due to propagation patterns or interferences. Therefore future versions of the routing protocol must be able to handle asymmetric links.

Dynamic packet configuration may be used to improve route throughput. The use of asymmetric data packets, e.g., DH5 in one direction and DH1 in the other, results in lower throughput for the DH1 direction.

Scatternet synchronization is needed to improve route throughput. The use of scatternet scheduling algorithms will synchronize data transmissions along the route hops. Thereby the situation of a master transmitting to a slave that is participating in another piconet at that time, can be avoided.

Finally, the implemented protocol does not use link metrics and assumes that the best path is the one that is shortest in time. In future developments a metric providing information of the link quality may improve route throughput.

- **Zigbee**

Distance estimation with Zigbee has been explored in Chapter 6, however the implementation of local positioning with Zigbee has not been described. The connection less broadcasting nature of Zigbee may be used to report LQI information to surrounding devices. This fact combined with distributed ad-hoc positioning algorithms may provide fast positioning information.

Soft handover algorithms rely on route discovery times as described in section 6.5.2. If the route path discovery algorithm is initiated in advance by the new "AP" node, handover latencies may be reduced. One limitation of the developed algorithm is that it can not be applied to RFDs, therefore the design of alternative association processes is needed. These new association processes will allow changes of parent for an end-device.

The developed hard handover algorithm may be used to improve link quality, if a PAN channel presents interference, the device may join a new PAN on a different channel. Moreover it could be desirable that the whole PAN uses adaptive channel election, in this case a new protocol must be designed to guarantee QoS while the network is reconfiguring the channel.

GTS can not be used in mesh networks, therefore synchronous communications are not allowed in mesh multi-hop networks. The design of multi-hop
routing protocols supporting GTS will enable synchronous communications in mesh networks.
Bibliography


