Chapter 2

Literature Study

This chapter overviews the studied literature and provides some background to the research area. Special attention is given to the IEEE 802.15.4 standard, routing in WSNs, existing WSN testbeds as well as existing WSN hardware.

2.1 Wireless Sensor Networks (WSNs)

A WSN is a collection of independent sensors that form a cooperative network [12]. Data leaves the network via one or more sink nodes. These sink nodes act as network gateways through which all the nodes in the network can be accessed. Not all the nodes within the network can communicate with a sink node directly. This is due to their limited communication range. These nodes pass data to a sink node on a hop-by-hop basis using other sensor nodes [13]. A WSN can consist of hundreds and even thousands of nodes. WSNs are used in a wide array of applications including home automation, building automation, personal healthcare, industrial control, utility meter communication and battlefield surveillance [1], [2]. The energy consumption of a WSN is affected by, amongst other things, the routing protocol employed in the network.
Figure 2.1 depicts a typical wireless sensor network. WSNs can be divided into two groups: proactive WSNs and reactive WSNs.

![Figure 2.1: A Depiction of a Typical Wireless Sensor Network](image)

2.1.1 Proactive WSN

The nodes that form a proactive WSN periodically sense the physical environment and transmit the sensed data to a sink node or base station. Proactive WSNs are suitable when periodic data measurement is required [6]. Nodes in such an WSN are capable of using a sleep mode to conserve energy. Figure 2.2 depicts a typical proactive WSN.
2.1.2 Reactive WSN

Nodes in a reactive WSN respond to a certain environmental event or stimulus by forwarding data about the event or stimulus to a sink node or base station [6]. Due to the possible irregularity of environmental stimuli, it is harder for nodes in a reactive WSN to make use of a sleep mode to conserve power. Figure 2.3 depicts a typical reactive WSN.

Request Driven WSN

A request driven WSN is a form of reactive WSN where the stimulus which prompts a node to forward data to a sink node or base station is a data request from a sink node.
or base station. Figure 2.4 depicts a typical request driven WSN.

2.1.3 Static and Mobile WSNs

In recent years commercial interest has shifted from small networks consisting of mobile nodes to large networks of static nodes. These networks include: convergecast WSNs in urban, home, building and industrial application areas. This has an effect on the routing solution chosen for a WSN and is reflected in the Internet Engineering Task Force (IETF) Routing Over Low Power and Lossy Networks (ROLL) standard [14].
Chapter 2 OSI Model

2.2 OSI Model

The Open Systems Interconnection (OSI) committee is a subcommittee of the International Organization for Standardization (IOS). The OSI committee was founded in 1977 and put forward a seven layer framework which is used to model distributed communication systems. This model was established due to the need for standards governing heterogeneous computer networks and functions as a framework guiding the definition of these standards [15].

Figure 2.4: A Depiction of a Request Driven Wireless Sensor Network
### 2.2.1 OSI Model Layers

This section provides an overview of the different OSI model layers based on the layer overview provided in [15]. The different layers of the OSI model can be seen in Table 2.1.

#### Physical Layer

The physical layer defines how the physical links between nodes are established, maintained and released. The different mechanical, electrical and procedural specifications of these links are provided by this layer. The physical layer therefore defines how devices interface with a transmission medium and also manages the transmission of data over the transmission medium [15].

#### Data Link Layer

The data link layer defines the functional and procedural specifications that govern the establishment, maintenance and release of data links between nodes [15]. The detection and correction of errors that originated in the physical layer also forms part of the duties performed by the data link layer. The data link layer is responsible for moving data between adjacent nodes [16].
Network Layer

The network layer provides the functional and procedural means which ensure the source to destination delivery of network service data units [15]. This includes the delivery of packets across multiple networks. Other important duties of this layer include logical addressing and routing [16].

Transport Layer

The transport layer provides a complete transport service in union with the physical, data link and network layers [15]. This layer is responsible for the delivery of data between processes running on different devices. This layer is also responsible for the segmentation and reassembly of a message [16].

Session Layer

The session layer forms relationships between entities. This layer is responsible for binding entities into relationships and unbinding them when communication is finished. This layer adds synchronization to data transfers [15].

Presentation Layer

The presentation layer helps the application layer to interpret the meaning of exchanged data [15]. The purpose of the presentation layer is to translate, encrypt and compress the data being exchanged [16].

Application Layer

The application layer exists to serve the end user by providing the distributed information service required by certain applications. The other layers of the OSI model exist to
support this layer [15].

2.3 Communication Protocol Standards

This section outlines some of the current standards that are applicable to WSNs. Special attention is given to the IEEE 802.15.4 standard.

2.3.1 IEEE Standard 802.15.4

The IEEE 802.15.4 standard defines a physical communication layer and a data link communication layer for Low-Rate Wireless Personal Area Networks (LR-WPANs) with specific focus on low power consumption [17]. The Medium Access Control (MAC) and Physical Layer (PHY) specifications of this standard are defined in IEEE Std. 802.15.4-2003. The standard utilizes Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA). Star and peer-to-peer network topologies are supported.

IEEE 802.15.4 Physical Layer (PHY)

The PHY consists of two services: the PHY data service and the PHY management service. The PHY management service interfaces with the MAC sublayer using the Physical Layer Management Entity (PLME) Service Access Point (SAP). The PHY data service accomplishes the transmission and reception of PHY Protocol Data Units (PPDUs) across the physical medium. PHY duties include: radio transceiver activation and deactivation, Energy Detect (ED) management, Link Quality Indicator (LQI) management, channel selection, Clear Channel Assessment (CCA), and the transmission as well as reception of packets over the physical medium [17]. The size of the PHY Maximum Transmission Unit (MTU) is 127 bytes. An IEEE 802.15.4 compliant transceiver operates in the following bands [17]:
• 868 – 868.6 MHz (e.g., Europe);
• 902 – 928 MHz (e.g., North America);
• 2400 – 2483.5 MHz (worldwide).

Please refer to [17] for the PHY specifications and regulatory requirements. Table 2.2 depicts the frequency band, modulation and data rate details of the PHY [17].

<table>
<thead>
<tr>
<th>PHY (MHz)</th>
<th>Frequency band (MHz)</th>
<th>Modulation</th>
<th>Bit rate (kb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>868/915</td>
<td>868 – 868.6</td>
<td>BPSK</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>902 – 928</td>
<td>BPSK</td>
<td>40</td>
</tr>
<tr>
<td>868/915 (optional)</td>
<td>868 – 868.6</td>
<td>ASK</td>
<td>250</td>
</tr>
<tr>
<td></td>
<td>902 – 928</td>
<td>ASK</td>
<td>250</td>
</tr>
<tr>
<td>868/915 (optional)</td>
<td>868 – 868.6</td>
<td>O-QPSK</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>902 – 928</td>
<td>O-QPSK</td>
<td>250</td>
</tr>
<tr>
<td>2450</td>
<td>2400 – 2483.5</td>
<td>O-QPSK</td>
<td>250</td>
</tr>
</tbody>
</table>

IEEE 802.15.4 MAC Sublayer

The MAC Sublayer consists of two services: the MAC data service and the MAC management service. The MAC sublayer interfaces with the upper layer using the MAC Sublayer Management Entity (MLME) SAP. The transmission and reception of MAC Protocol Data Units (MPDUs) is accomplished by the MAC data service. The duties of the MAC sublayer include: beacon management, channel access, Guaranteed Time Slot (GTS) management, frame validation, acknowledged frame delivery, association, and disassociation. The size of the MAC MTU is 118 bytes [17].

IEEE 802.15.4 Frame Structures

The IEEE 802.15.4 Standard defines four frame structures which were designed to reduce complexity while still providing a robust solution for noisy channels. These frames include:
• A beacon frame;
• A data frame;
• An acknowledgement frame;
• A MAC command frame.

The structure of the beacon frame is illustrated in Figure 2.5. Network coordinators broadcast beacon frames when beacon frames are enabled in the Personal Area Network (PAN). The MAC sublayer header consists of the MAC frame control field, beacon sequence number (BSN), addressing fields, and the optional auxiliary security header. The MAC payload contains the superframe specification, GTS fields, pending address fields, and beacon payload. The MAC footer contains a 16-bit Frame Check Sequence (FCS) [17].

![](image)

Figure 2.5: A Diagram of the IEEE 802.15.4 Beacon Frame and Physical Layer Packet [17]

The data frame is presented in Figure 2.6. Data frames are used to facilitate data transfers. The MAC sublayer header consists of the frame control field, Data Sequence Number (DSN), addressing fields, and the optional auxiliary security header. The MAC payload contains the data transferred to the MAC sublayer from upper layers. The MAC footer contains a 16-bit FCS [17].

The acknowledgement frame is depicted in Figure 2.7. Acknowledgement frames are used to confirm successful frame receptions. This frame has no payload. The MAC sublayer header consists of the MAC frame control field and DSN. The MAC footer contains a 16-bit FCS [17].
The structure of the command frame is illustrated in Figure 2.8. Command frames facilitate MAC peer entity control transfers. The MAC sublayer header consists of the MAC frame control field, DSN, addressing fields, and the optional auxiliary security header. The MAC payload contains the command type field and the command payload. The MAC footer contains a 16-bit frame check sequence (FCS) [17].
2.3.2 Microchip Wireless (MiWi) Media Access Controller (MiMAC)

The Microchip Wireless (MiWi) Media Access Controller (MiMAC) defines a MAC and PHY which are compatible with all Microchip transceivers across different frequency bands and modulations. When MiMAC is used in conjunction with a MRF24J40 transceiver, the combination complies with the IEEE 802.15.4 standard [18].

MiMAC Frame Structure

The MiMAC layer consists of three separate modules. These modules include the MiMAC frame format, MiMAC security module and the MiMAC universal programming interface. The MiMAC frame format defines how the message appears on the physical medium and forms a basis for the other MiMAC modules. The MiMAC frame structure can be seen in Figure 2.9. The MiMAC security module implements a low-cost block cipher. The MiMAC universal programming interface forms a convenient interface between all Microchip RF transceivers and Microchip proprietary wireless communication protocols [18].

2.3.3 ZigBee

The Zigbee standard is a network layer standard that builds on the IEEE 802.15.4 standard. The standard is designed to be an energy efficient and low cost short range wireless communication standard. The routing protocol used for the ZigBee standard is based the Ad Hoc On-Demand Distance Vector (AODV) routing protocol. Nodes within a ZigBee network perform different functions depending on whether the node is a ZigBee Coordinator (ZC), ZigBee Router (ZR) or ZigBee End Device (ZED). A coordination device assigns addresses to the other devices in the network. A routing device can cluster end devices under itself [19]. Figure 2.10 depicts a typical ZigBee based WSN.
2.3.4 Internet Engineering Task Force (IETF) Routing Over Low Power and Lossy Networks (ROLL)

The IETF ROLL working group was created in 2008 to standardize routing in low power and lossy networks. Presently the IETF ROLL working group is in the final stages of completing a recommendation for these networks. The solution is based on self-organizing coordinate systems. Commercial interest in WSNs has moved into the direction of large static networks. The IETF ROLL recommendation takes this into account [14].

2.4 Routing Protocols

No one routing protocol can deliver optimal performance in all applications. Selecting an appropriate routing protocol for a wireless sensor network depends largely on the
specific requirements of the network [20].

### 2.4.1 Proactive Routing Protocols

Proactive routing protocols acquire and maintain information about the network topology and link states even if all this information is not needed. This applies to every node within the network. The routing tables of these nodes are kept up to date by flooding the network with request messages.
Open Shortest Path First (OSPF)

Open Shortest Path First (OSPF) is an open source link-state routing protocol which was designed to function inside an Autonomous System (AS). Each router running this protocol maintains a database describing the network topology. The path to a destination is calculated by constructing a shortest-path tree using this database. OSPF supports equal-cost multipath load balancing [21].

Optimized Link State Routing (OLSR)

The Optimized Link State Routing (OLSR) protocol is an adaptation of the classical link state algorithm which was developed to suite the needs of Mobile Ad Hoc Networks (MANETs). Link state routing protocols converge quickly and are less prone to routing loops than distance vector routing protocols. However link state routing protocols do require more memory and processing power than distance vector routing protocols. OLSR uses multipoint relays (MPRs) to control the forwarding of broadcast messages during the flooding process. This reduces the energy consumption of the network [22].

2.4.2 Reactive Routing Protocols

Reactive routing protocols build routes on demand. This means that routing table updates are kept to a minimum which saves energy in power constrained networks. These protocols do not perform well in networks which require frequent data exchanges. This is because the amount of control messages necessary to set up routes would consume all bandwidth and drain the network energy.

Ad Hoc On-Demand Distance Vector (AODV)

This routing protocol was designed specifically for ad-hoc networks. Each device in the network acts as a router which enables the network to function without requiring pre-
existing infrastructure [23]. A node using this routing protocol only acquires the path to a destination device when it has data ready to be sent. This means that the routing overhead of this protocol can become problematic in large networks. The protocol also heavily depletes the energy resources of a small subset of nodes which form part of the shortest-hop path due to overuse [24].

**Multipath Energy Aware AODV Routing (ME-AODV)**

ME-AODV is an energy aware wireless sensor network routing protocol that outperforms the AODV routing protocol due to the fact that it decreases routing overhead and increases the network lifetime. The protocol forms logical clusters within the network. The protocol increases the network lifetime by hopping between different paths, between communicating nodes, in a round robin fashion. This means that the nodes on the shortest-hop path are not overused [24].

### 2.4.3 Hierarchical Routing Protocols

These routing protocols organize a network in a hierarchical manner and form clusters within the network. This is accomplished by assigning different functions to some nodes within the network. This gives rise to nodes or devices that are specialized to perform certain functions. Endpoint nodes focus on data collection functions, while other nodes function as cluster-head nodes. These cluster-head nodes cluster nodes under themselves and focus on performing routing functions for these clustered nodes. Gateway nodes act as network gateways. Some hierarchical routing protocols are detailed in this section.

**Low Energy Adaptive Clustering Hierarchy (LEACH)**

The Low Energy Adaptive Clustering Hierarchy (LEACH) protocol is an integration of a Time Division Multiple Access (TDMA) based MAC protocol, clustering and a
simple routing protocol which was designed for WSNs. This protocol functions in a completely distributed manner and does not require any management by a base station. Some limitations of this protocol include [25]:

- Cluster-heads tend to get overused;
- Cluster-heads are predefined and require a uniform distribution;
- The remaining energy of a node is not taken into account during cluster-head selection;
- There is a significant variation in cluster sizes;
- A lot of energy is consumed during each round when new clusters are formed.

Threshold Sensitive Energy Efficient Sensor Network Protocol (TEEN)

The Threshold Sensitive Energy Efficient Sensor Network Protocol (TEEN) routing protocol was specifically designed for reactive WSNs. This protocol ensures that time critical data reaches the user quickly. Because periodic routing table updates are not required the protocol conserves network energy and bandwidth [6].

2.5 Energy Aware Routing Schemes

Due to the energy constraints of WSNs energy aware routing schemes play an important role in maximizing the network lifetime. In this section some energy aware routing schemes are outlined.

2.5.1 Minimum Total Transmission Power Routing (MTTPR)

MTTPR chooses the path with the lowest total transmission power to forward data between communicating nodes. According to previous simulation results mobility im-
pairs the performance of this protocol [26]. Because the remaining energy of each node is not taken into account some nodes within the network may get overused. The total transmission power $P_j$ for route $j$ is given by Equation 2.1. $P(n_i, n_{i+1})$ is the transmission power used between node $n_i$ and node $n_{i+1}$. $D$ is the hop distance between the source node ($n_0$) and destination node ($n_D$) [27].

$$P_j = \sum_{i=0}^{D-1} P(n_i, n_{i+1})$$  \hspace{1cm} (2.1)

The Minimum Total Transmission Power (MTTP) route $m$ is selected using Equation 2.2. $A$ is a set containing all the possible routes.

$$P_m = \min A = \min\{P_0, P_1, P_2, ..., P_j\}$$  \hspace{1cm} (2.2)

### 2.5.2 Minimal-Battery Cost Routing (MBCR)

The Minimal-Battery Cost Routing (MBCR) scheme calculates the total remaining battery life along each path from the data source node to the destination node and chooses the path which has the most battery life remaining in total [28]. This scheme does not take the remaining battery life of single nodes into account. Therefore single nodes may get overused. Let $b_i$ represent the remaining battery capacity of node $n_i$, ranging from 0 to 100. The function $f_i$ is used to represent the willingness of a node to forward data based on its remaining battery capacity. Division by 0 is not possible because a value of 0 would mean that the node is not powered.

$$f_i(b_i) = \frac{1}{b_i}$$  \hspace{1cm} (2.3)

The total battery cost $B_k$ for route $k$ is given by Equation 2.4. $D$ is the hop distance between the source and destination nodes.
Chapter 2  
Energy Aware Routing Schemes

\[ B_k = \sum_{i=0}^{D-1} f_i(b_i) \]  

(2.4)

The maximum total remaining battery capacity route \( z \) is selected using Equation 2.5. \( A \) is a set containing all the possible routes. When two or more routes have the same (lowest) cost the route with the lowest number of hops is selected.

\[ B_z = \min A = \min \{B_0, B_1, B_2, ..., B_k\} \]  

(2.5)

2.5.3 Min-Max Battery Cost Routing (MMBCR)

Min-Max Battery Cost Routing (MMBCR) avoids routes which contain nodes with the lowest remaining battery life. The MMBCR routing scheme therefore ensures that no nodes within the network get overused, however, it can not guarantee that the chosen route is a MTTPR route or a MBCR route. The cost of the route \( k \) as given by Equation 2.4 is rewritten in Equation 2.6.

\[ B_k = \max_{i \in k} f_i(b_i) \]  

(2.6)

The min-max battery cost route \( s \) is selected using Equation 2.7. \( A \) is a set containing all the possible routes [27]. When two or more routes have the same (lowest) cost the route with the lowest number of hops is selected.

\[ B_s = \min A = \min \{B_0, B_1, B_2, ..., B_k\} \]  

(2.7)
2.5.4 **Conditional Max-Min Battery Capacity Routing (CMMBCR)**

Conditional Max-Min Battery Capacity Routing (CMMBCR) combines the benefits of MTTPR and MMBCR. MTTPR is used to choose routes, however if the remaining energy of a node on the route chosen by MTTPR is below a certain threshold level (\( \gamma \)) another route is chosen. This means that single nodes are not overused [27].

2.5.5 **Minimum Total TransCeiving Power (MTTCP)**

The Minimum Total TransCeiving Power (MTTCP) routing scheme is based on the MTTP routing scheme. The difference between the two protocols is that the MTTCP routing scheme tries to minimize the total transmission and reception power used on the path between a source node and a destination node [29].

2.5.6 **Minimum Total Reliable Transmission Power (MTRTP)**

Minimum Total Reliable Transmission Power (MTRTP) routing is based on the MTTP routing scheme. The MTRTP routing scheme takes the reliability as well as the transmission power setting of each link into account [29].

2.6 **Routing Metrics**

Routing metrics are used by a routing protocol to select the most appropriate path between communicating nodes. These metrics are used to describe the throughput, bandwidth, delay, hop count and total transmission power of a path to name but a few. This section provides some detail on commonly used routing metrics.
2.6.1 Hop Count

The hop count metric provides the number of hops a message must undergo to reach a certain destination node. The hops that are counted are between intermediate devices which function on the OSI network layer.

2.6.2 Expected Transmission Count (ETX)

This metric minimizes the expected total number of transmission required to deliver a message. The Expected Transmission Count (ETX) metric takes the link loss ratio, link asymmetry and interference among successive links on each path into account [30].

2.6.3 Energy Aware Routing Metrics

These metrics are used to implement different energy aware routing schemes including MTTPR, MTTCP, MTRTP, MBCR, MMBCR and CMMBCR. These metrics can also be used in combination with other metrics to form new composite metrics which take the energy consumption of the network into account.

2.7 WSN Testbed Architectures

This section details the TWIST and WISEBED WSN testbed architectures.

2.7.1 WISEBED

WISEBED is a joint effort between nine different academic institutions across Europe, which is funded by the European Commission under the Information Communication Technologies programme. This architecture aims to create well organised large scale
WSNs consisting of heterogeneous nodes. The architecture integrates the management of network hardware, software, algorithms, and data. The WISEBED hardware architecture is depicted in Figure 2.11. Some SNs connect to network gateways via USB. Network gateways are usually embedded PCs or netbook-class computers. These gateways connect to the testbed server using an Ethernet backbone. The testbed client accesses the network using the testbed server. The SNs used in WISEBED testbeds are: Pacemate, iSense, TelosB, Tmote, MicaZ, SunSPOT, MSB-A2, Tmote Sky and G-Node [31].

![Figure 2.11: WISEBED Hardware Architecture Diagram](image)

**2.7.2 TKN Wireless Indoor Sensor network Testbed (TWIST)**

The TKN Wireless Indoor Sensor network Testbed (TWIST) architecture is a scalable and adaptable WSN test-bed architecture developed by the Telecommunication Net-
works Group (TKN) at the Technische Universität Berlin. This testbed architecture supports the basic features required in a WSN testbed which includes: node configuration, network-wide programming, out-of-band extraction of debugging data and application data gathering. The architecture also includes novel features like: heterogeneous node platform support, power supply control (which allows network reconfiguration and node failure injection) and the creation of flat and hierarchical networks using “super nodes”. The nodes used in the TKN testbed are eyesIFX and Telos nodes. The TWIST hardware architecture is depicted in Figure 2.12. SNs connect to super nodes using active or passive USB connections and a USB hub. These super nodes in turn connect to the testbed server and control station using an Ethernet backbone [32].

Figure 2.12: TWIST Hardware Architecture Diagram [32]
2.8 Existing WSN Testbeds

This section details some existing WSN testbeds (excluding those covered in the previous section) as well as SN power consumption measurement in WSN Testbeds. In 2011, Steyn and Hancke surveyed WSN testbeds [33]. Hellbrück et al. also surveyed publicly available WSN testbeds as well as the SNs used in WISEBED based WSN testbeds in 2011 [31]. Table 2.3 lists some features of existing testbeds as provided by Steyn and Hancke [33] as well as Hellbrück et al. [31]. The rest of this section provides some more detail on selected WSN testbeds as well as power consumption measurement in WSN testbeds.

2.8.1 Senslab

Senslab is a very large scale open source WSN. Senslab consists of 1024 WSN430 nodes spread across four sites. Each site has 256 SNs available. Two sites use an IEEE 802.15.4 radio interface while the remaining sites use IEEE 802.11 and 868 MHz Free MAC Layer radio interfaces. The operating systems used on the testbed SNs include: Contiki, FreeRTOS, TinyOS and RIOT. The testbed SNs can be configured remotely and can be used without any restrictions in terms of Operating System (OS), programming language, drivers or Application Programming Interface (API). SNs can be configured as sink nodes and mobile nodes can be incorporated into experiments [34].

2.8.2 LabVIEW TB

This LabVIEW based testbed consists of 23 SNs and was developed to experiment with mobile SNs. The SNs used are MicaZ and Cricket nodes. Some nodes in the testbed are static while others have been deployed on mobile platforms and humans. All nodes in the network are programmed and controlled using the LabVIEW programming environment. TinyOS is used on the SNs and the LabVIEW user interface is able to install
### Chapter 2 Existing WSN Testbeds

#### Table 2.3: Existing Testbed Sensor Nodes

<table>
<thead>
<tr>
<th>Testbed</th>
<th>Year</th>
<th>Sensor Node Types</th>
<th>Number of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>WISEBED</td>
<td>2011</td>
<td>Pacemate, iSense, TelosB, Tmote, MicaZ, SunSPOT, MSB-A2, Tmote Sky, G-Node</td>
<td>738</td>
</tr>
<tr>
<td>Motelab</td>
<td>2005</td>
<td>MicaZ, Tmote Sky</td>
<td>190</td>
</tr>
<tr>
<td>Kansei</td>
<td>2006</td>
<td>XSM, Tmote Sky</td>
<td>190</td>
</tr>
<tr>
<td>LabVIEW TB</td>
<td>2006</td>
<td>MicaZ, Cricket</td>
<td>23</td>
</tr>
<tr>
<td>SignetLab</td>
<td>2007</td>
<td>EyesIFXv2</td>
<td>48</td>
</tr>
<tr>
<td>TWIST</td>
<td>2006</td>
<td>TelosB, EyesIFX</td>
<td>180</td>
</tr>
<tr>
<td>MoteMaster</td>
<td>2009</td>
<td>TelosB</td>
<td>-</td>
</tr>
<tr>
<td>X-Sensor</td>
<td>2009</td>
<td>MicaZ</td>
<td>178+</td>
</tr>
<tr>
<td>Senslab</td>
<td>2010</td>
<td>WSN430</td>
<td>1000+</td>
</tr>
<tr>
<td>HINT</td>
<td>2010</td>
<td>Modified TelosB</td>
<td>-</td>
</tr>
</tbody>
</table>

TinyOS programs [35].

#### 2.8.3 MoteMaster

The MoteMaster testbed was developed by the Department of Wireless Networks at Aachen University. This testbed consists of TelosB nodes. SNs are connected to mote managers (PCs) using USB hubs. These mote managers in turn connect to the testbed backbone (database, file server and dispatcher). The testbed is managed using a Command and Control Scripting Language (CCL) which was designed specifically for the testbed. MATLAB and legacy C Sharp interfaces are available for network management [36].

#### 2.8.4 HINT

HINT is a nonintrusive and high accuracy testbed. This testbed is nonintrusive because the different testbed operations, including data collection, do not disturb the spontaneous behaviour of the network while an experiment is being conducted. The HINT testbed is comprised of ZiNT nodes, which are essentially TelosB nodes with more pins available on the expansion slots. These extra pins are used in conjunction with
Field-Programmable Gate Arrays (FPGAs) to monitor, amongst other things, the node power consumption. The SNs connect to the testbed server via an ethernet backbone [37].

### 2.8.5 Sensor Node Power Consumption Measurement in WSN Testbeds

Most WSN testbeds use built in power consumption metrics to estimate power consumption. Other testbed nodes use specialised external hardware to provide high accuracy energy consumption data [33], [38], [39]. The results of routing experiments conducted using energy consumption ascertaining WSN testbeds are typically presented in terms of the total energy consumption of the network, the energy consumption of the different network components and the node expiry sequence.

### 2.9 Existing Sensor Node Hardware

This section details some existing wireless sensor nodes that are used in WSN testbeds. Commonly used WSN testbed SNs include: MICAz, SunSPOT, iSense, TelosB, G-Node and Tmote Sky. Some features of existing SNs are provided in Table 2.4 [40], [41], [42], [43], [44], [45], [46], [47], [48], [49]. The microcontrollers that are used in SN designs range from low power 8-bit microcontrollers to powerful 32-bit ARM processors. When selecting a microcontroller for a SN application a compromise has to be made between its power consumption and processing capabilities. Most SNs are powered by two AA (R6) batteries. These nodes do not have a battery charger incorporated into the design. This is because most SNs will not be redeployed during their lifetime in various applications [50]. The majority of existing sensor nodes make use of an IEEE standard 802.15.4 compliant transceiver.
Table 2.4: Existing Sensor Node Hardware

<table>
<thead>
<tr>
<th>Device</th>
<th>Program Memory</th>
<th>Measurement Memory</th>
<th>IEEE 802.15.4</th>
<th>Battery Capacity</th>
<th>Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICAz</td>
<td>128kB</td>
<td>512kB</td>
<td>Yes</td>
<td>2 AA</td>
<td>UART, GPIO, I2C, 10 bit A/D</td>
</tr>
<tr>
<td>Tinynode 584</td>
<td>48kB</td>
<td>512kB</td>
<td>No</td>
<td>2/3 AA</td>
<td>UART, SPI, GPIO, 12-bit A/D, 12-bit D/A</td>
</tr>
<tr>
<td>BTnode</td>
<td>128kB</td>
<td>4kB</td>
<td>No</td>
<td>2 AA</td>
<td>UART, SPI, I2C, GPIO, A/D</td>
</tr>
<tr>
<td>Tmote Sky</td>
<td>48kB</td>
<td>1024kB</td>
<td>Yes</td>
<td>2 AA</td>
<td>UART, USB, I2C, GPIO, 12-bit A/D, 12-bit D/A</td>
</tr>
<tr>
<td>LOTUS</td>
<td>512kB</td>
<td>64MB</td>
<td>Yes</td>
<td>2 AA</td>
<td>USB, A/D, GPIO, I2C, SPI, UART</td>
</tr>
<tr>
<td>iSense</td>
<td>512kB</td>
<td>ADJ</td>
<td>Yes</td>
<td>ADJ</td>
<td>USB, UART, SPI, GPIO, I2C, 12-bit A/D, 12-bit D/A</td>
</tr>
<tr>
<td>TelosB</td>
<td>48kB</td>
<td>1MB</td>
<td>Yes</td>
<td>2 AA</td>
<td>USB, UART, SPI, GPIO, I2C, 12-bit A/D, 12-bit D/A</td>
</tr>
<tr>
<td>SunSPOT</td>
<td>8MB</td>
<td>1MB</td>
<td>Yes</td>
<td>Li-Ion 770mAhr</td>
<td>USB, UART, SPI, GPIO, I2C, 12-bit A/D, 12-bit D/A</td>
</tr>
<tr>
<td>G-Node</td>
<td>116kB</td>
<td>8Mbit</td>
<td>Yes</td>
<td>none</td>
<td>USB, UART, SPI, GPIO, I2C, 12-bit A/D, 12-bit D/A</td>
</tr>
<tr>
<td>MSB-A2</td>
<td>512kB</td>
<td>SDcard</td>
<td>Yes</td>
<td>none</td>
<td>USB, UART, SPI, GPIO, RS232, I2C, 10-bit A/D, 10-bit D/A</td>
</tr>
</tbody>
</table>

2.9.1 MICAz

The MICAz wireless measurement system is produced by Crossbow Technology, Inc. These sensor nodes are supported by the MoteWorks WSN platform. MoteWorks was developed to facilitate ad hoc mesh networking in low-power, battery-operated networks. The MICAz system is based on a 8-bit Atmel ATmega128L microcontroller and...
has 128 kB of flash program memory and 512 kB of serial flash memory for measurement purposes. The available interfaces include: UART, Digital I/O, I2C and SPI. Each node is powered by two AA batteries. The MICAz sensor node is ZigBee compliant and contains an IEEE 802.15.4 compliant transceiver [46].

2.9.2 Tmote Sky

The Tmote Sky wireless sensor node is based on an 8 MHz Texas Instruments MSP430 microcontroller. Each node has 48 kB of program memory and 1024 kB of flash measurement memory available. Each Tmote Sky node has an USB port and an IEEE standard 802.15.4 compliant transceiver. The Tmote Sky comes with out-of-the-box TinyOS support. On-board humidity, light and temperature sensors are available [45].

2.9.3 Tinynode 584

The Tinynode 584 sensor node is produced by the Shockfish company. This node was designed and optimized to run the TinyOS and comes with nineteen configurable I/O pins. The node comes with a 2.5 ppm TCXO RF clock reference. The node is available in three different versions depending on the required ISM band operation. 433 MHz, 868 MHz and 915 MHz ISM band nodes are available [40].

2.9.4 BTnode

The BTnode is a test platform for distributed wireless sensor networks which was developed at ETH Zurich by the Computer Engineering and Networks Laboratory (TIK) and the Research Group for Distributed Systems. The node is based on a 8-bit Atmel ATmega 128L microcontroller and has 128 kB of flash program memory. Each node has 4 kB of EEPROM for measurement purposes. Each BTnode node has two transceivers. The first is a Zeevo ZV4002 Bluetooth transceiver and the second is a Chipcon CC1000
Existing Sensor Node Hardware

Chapter 2

radio operating in the 433-915 MHz ISM band. The BTnode system can be used with TinyOS [41].

2.9.5 LOTUS

The LOTUS SN is an advanced wireless sensor platform which is built around the 32-bit ARM Cortex M3 processor. The LOTUS node is Zigbee and IEEE standard 802.15.4 compliant. Each node has a on-board antenna with a nominal communication range of 100 m. LOTUS nodes have 512 kB of program memory and 64 MB of measurement memory available [44].

2.9.6 iSense

The iSense WSN hardware modules by Coalesenses are designed for research and industrial applications. Each iSense node consists of a core module which can be fitted with various sensor modules or custom modules. Many energy module options are also available. These include solar and battery power modules. The core module is based on a 32-bit RISC microcontroller which has 128 kB of RAM and 512 kB of Flash memory. The iSense’s transceiver is IEEE standard 802.15.4 compliant and Zigbee ready [49].

2.9.7 SunSPOT

The Sun SPOT node is produced by Sun Microsystems Incorporated. Each node consists of a main processor board (eSPOT) and an application board. The eSPOT runs the Java Squawk virtual machine. An IEEE standard 802.15.4 compliant CC2420 transceiver is used on the board. The main processor is an 400 MHz ARM 926ej-S Processor. Each node has 8 MB of flash memory and 1 MB SRAM memory available to the user. A full speed USB 2.0 interface is available for programming and serial communication. The
node can be powered by a rechargeable 770 mAh Li-Ion battery or via the USB inter-
face. The eSPOT is capable of measuring its current consumption and battery voltage.
The power supply output voltage is not measured [48].

2.10 Operating Systems

This section outlines some of the currently available operating systems which can be
used in WSN applications.

2.10.1 TinyOS

TinyOS is an open source operating system which was specifically designed for wire-
less devices with low power requirements. The TinyOS operating system is BSD-
licensed. TinyOS currently supports the following microcontrollers [51]:

- Atmel ATmega128;
- Texas Instruments MSP430;
- Intel XScale PXA271.

Some ports to other microcontrollers are available.

2.10.2 FreeRTOS

FreeRTOS is an open source real time operating system which is produced by Real
Time Engineers Ltd. This operating system is very well documented, supported and
strictly quality controlled. FreeRTOS currently supports various microcontroller fami-
lies from many vendors including [52]:


• Altera;
• Atmel;
• Fujitsu;
• Silicon Labs;
• Texas Instruments.
2.11 Conclusion

This chapter detailed some literature relevant to the research area. The chapter focused on routing in WSNs, applicable standards, existing WSN testbeds and existing WSN hardware. In the next chapter we present the design of an energy consumption ascertaining WSN testbed SN and provide some motivation for the development of this SN.