<table>
<thead>
<tr>
<th>Title</th>
<th>Evaluation models for different routing protocols in wireless sensor networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors(s)</td>
<td>Cao, Ning</td>
</tr>
<tr>
<td>Publication date</td>
<td>2015</td>
</tr>
<tr>
<td>Publisher</td>
<td>University College Dublin. School of Computer Science</td>
</tr>
<tr>
<td>Link to online version</td>
<td><a href="http://dissertations.umi.com/ucd:10079">http://dissertations.umi.com/ucd:10079</a></td>
</tr>
<tr>
<td>Item record/more information</td>
<td><a href="http://hdl.handle.net/10197/8538">http://hdl.handle.net/10197/8538</a></td>
</tr>
</tbody>
</table>

Downloaded 2018-12-12T17:56:37Z

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

Some rights reserved. For more information, please see the item record link above.

Ning Cao

The thesis is submitted to University College Dublin in fulfilment of the requirement for the degree of PhD in the College of Science.

School of Computer Science

Head of School: Professor P. Cunningham

Principal Supervisors: Professor G.M.P. O'Hare
Dr R. Higgs

November 2015
Acknowledgments

This research is based upon works supported by the Clarity Centre, University College Dublin.
Publications to date


Contents

Chapter 1: Introduction ............................................................................................................. 1
  1.1. Introduction ...................................................................................................................... 1
  1.2. Motivation ........................................................................................................................ 2
  1.3. Objectives ........................................................................................................................ 3
  1.4. Contributions of the Thesis ............................................................................................. 4
  1.5. Thesis Outline ................................................................................................................... 5

Chapter 2: Background ............................................................................................................... 6
  2.1. Wireless Sensor Networks ............................................................................................... 6
       2.1.1. Media Access Control ............................................................................................... 9
       2.1.2. Time-Division Multiple Access .................................................................................. 10
       2.1.3. Slotted ALOHA .......................................................................................................... 12
  2.2. Wireless Sensor Network Simulation Tools .................................................................... 12
       2.2.1. NS-2 .......................................................................................................................... 15
       2.2.2. OMNET++ .................................................................................................................. 16
       2.2.3. TOSSIM ...................................................................................................................... 17
       2.2.4. ATEMU ....................................................................................................................... 18
       2.2.5. EmStar ......................................................................................................................... 19
       2.2.6. J-Sim .......................................................................................................................... 20
       2.2.7. Comparison of Simulation Tools .................................................................................. 22
  2.3. Details of J-Sim .................................................................................................................. 23
       2.3.1. J-Sim Platform ............................................................................................................ 23
       2.3.2. Energy Consumption Models ...................................................................................... 23
       2.3.3. Reliability Models ....................................................................................................... 25
  2.4. Evaluation Parameters ...................................................................................................... 27
       2.4.1. Energy-efficiency Problem ......................................................................................... 27
Chapter 3: Routing Protocols

3.1. Flat Routing Protocols
   3.1.1. Single-hop Protocol
   3.1.2. Sensor Protocols for Information via Negotiation (SPIN)
   3.1.3. Directed Diffusion (DD)
   3.1.4. Rumor Routing (RR)
   3.1.5. Maximizing Energy Utilization Routing Protocol (MEURP)

3.2. Hierarchical Routing Protocols
   3.2.1. Threshold-sensitive Energy Efficient Protocol (TEEN)
   3.2.2. Low Energy Adaptive Clustering Hierarchy (LEACH)
   3.2.3. Cluster Based Routing Protocol (CBR)
   3.2.4. Quadrature-LEACH (Q-LEACH)
   3.2.5. Fixed-environment Location-based Clustering Routing Protocol
   3.2.6. Mixed-environment Location-based Clustering Routing Protocol

3.3. Location-based Routing Protocols
   3.3.1. Greedy Perimeter Stateless Routing (GPSR)
   3.3.2. Nearest with Forwarding Protocol (NFP)
   3.3.3. Nearest Closer Protocol (NC)
   3.3.4. Dijkstra-based Localized Energy-efficient Multicast Algorithm

3.4. Reasons for the selection of routing protocols
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4.2.</td>
<td>Radius—Lifetime—Reliability</td>
<td>84</td>
</tr>
<tr>
<td>6.4.3.</td>
<td>Cluster—Lifetime—Reliability</td>
<td>86</td>
</tr>
<tr>
<td>6.5.</td>
<td>Lifetime Model</td>
<td>91</td>
</tr>
<tr>
<td>6.6.</td>
<td>Reliability Model</td>
<td>95</td>
</tr>
<tr>
<td>6.7.</td>
<td>Summary</td>
<td>97</td>
</tr>
</tbody>
</table>

Chapter 7: Evaluation Models for the Nearest Closer Protocol ........................................ 98

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1.</td>
<td>System Architecture and Experimental Set-up</td>
<td>99</td>
</tr>
<tr>
<td>7.2.</td>
<td>Results</td>
<td>99</td>
</tr>
<tr>
<td>7.2.1.</td>
<td>Density—Lifetime—Reliability</td>
<td>99</td>
</tr>
<tr>
<td>7.2.2.</td>
<td>Radius—Lifetime—Reliability</td>
<td>104</td>
</tr>
<tr>
<td>7.3.</td>
<td>Lifetime Model</td>
<td>107</td>
</tr>
<tr>
<td>7.3.1.</td>
<td>Early Termination</td>
<td>120</td>
</tr>
<tr>
<td>7.3.2.</td>
<td>Data Build-Up</td>
<td>120</td>
</tr>
<tr>
<td>7.5.</td>
<td>Reliability Model</td>
<td>126</td>
</tr>
<tr>
<td>7.6.</td>
<td>Radius Model</td>
<td>131</td>
</tr>
<tr>
<td>7.7.</td>
<td>Summary</td>
<td>135</td>
</tr>
</tbody>
</table>

Chapter 8: Critique and Future Work ............................................................................. 137

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1.</td>
<td>Critique</td>
<td>137</td>
</tr>
<tr>
<td>8.1.1.</td>
<td>Routing Protocols</td>
<td>137</td>
</tr>
<tr>
<td>8.1.2.</td>
<td>Evaluation Parameters</td>
<td>138</td>
</tr>
<tr>
<td>8.1.3.</td>
<td>J-Sim and Simulation Results</td>
<td>139</td>
</tr>
<tr>
<td>8.1.4.</td>
<td>Equations</td>
<td>143</td>
</tr>
<tr>
<td>8.1.5.</td>
<td>Drawbacks of the approach</td>
<td>144</td>
</tr>
<tr>
<td>8.2.</td>
<td>Future Work</td>
<td>146</td>
</tr>
<tr>
<td>8.2.1.</td>
<td>Power Management</td>
<td>146</td>
</tr>
<tr>
<td>8.2.2.</td>
<td>The Construction of Evaluation Models for More Routing Protocols</td>
<td>148</td>
</tr>
<tr>
<td>8.2.3.</td>
<td>Mathematical Model Implementation</td>
<td>149</td>
</tr>
</tbody>
</table>
8.2.4. Other related work.................................................................................................................. 150

8.3. Summary.................................................................................................................................. 150

Chapter 9: Conclusions.................................................................................................................. 151

References........................................................................................................................................ 153
Abstract

This thesis aims to introduce the evaluation parameters of Lifetime, Density, Radius, and Reliability for the applications of wireless sensor networks. A series of simulation results have been obtained for the Single-hop, LEACH and Nearest Closer routing protocols which have been implemented in J-Sim simulation platform. Simulation results have been analyzed and several evaluation models have been proposed respectively. Thus, simulations may not be necessary for the users to choose a suitable routing protocol.
Statement of Original Authorship

I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree stated on the Title Page, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work.
LIST OF FIGURES

Figure 2.1: The Data Transmission Model of a Wireless Sensor Network ......................... 7
Figure 2.2: Simulation Model .......................................................................................... 14
Figure 2.3: OMNET++ Interface ...................................................................................... 16
Figure 2.4: TinyViz Interface .......................................................................................... 17
Figure 2.5: XATDB Interface .......................................................................................... 19
Figure 2.6: EmStar Interface .......................................................................................... 20
Figure 2.7: Sensor Node .................................................................................................. 21

Figure 3.1: Greedy Routing Strategies .............................................................................. 46

Figure 4.1: Simulation Area ............................................................................................. 55

Figure 5.1: Density, Reliability Relationship for Single-hop .............................................. 59
Figure 5.2: Density, Lifetime Relationship for Single-hop ................................................ 60
Figure 5.3: Radius, Reliability Relationship for Single-hop ................................................ 62
Figure 5.4: Radius, Lifetime Relationship for Single-hop ................................................ 63
Figure 5.5: A Model of the Reliability ............................................................................. 67
Figure 5.6: A Model of the Lifetime ............................................................................... 69

Figure 6.1: (a) Number of Sensors, Lifetime Relationship. (b) Number of Sensors, Reliability Relationship ................................................................. 77
Figure 6.2: Lifetime Plotted against Number of Sensors Times the Reliability ............... 81
Figure 6.3: (a) Number of Sensors, Lifetime Relationship. (b) Number of Sensors, Reliability Relationship ................................................................. 83
Figure 6.4: (a) Radius, Lifetime Relationship. (b) Radius, Reliability Relationship ....... 85
Figure 6.5: (a) Number of Clusters, Lifetime Relationship (20 Sensors). (b) Number of Clusters, Reliability Relationship (20 Sensors) ............................................. 87
Figure 6.6: Number of Clusters, Lifetime Relationship (30 Sensors). (b) Number of Clusters, Reliability Relationship (30 Sensors) ............................................. 88
Figure 6.7: Number of Clusters, Lifetime Relationship (40 Sensors). (b) Number of Clusters, Reliability Relationship (40 Sensors) ............................................. 90
Figure 6.8: A Model of the Lifetime ............................................................................... 93
Figure 6.9: A Model of the Reliability.................................................................96

Figure 7.1: Density, Reliability Relationship for NC..............................................100
Figure 7.2: Density, Lifetime Relationship for NC...............................................101
Figure 7.3: Radius, Reliability Relationship for NC..............................................104
Figure 7.4: Radius, Lifetime Relationship for NC...............................................105
Figure 7.5: Radius, Reliability Relationship for NC (20 Sensors)..........................106
Figure 7.6: Radius, Lifetime Relationship for NC (20 Sensors)..........................107
Figure 7.7: Typical Tree Zones...........................................................................110
Figure 7.8: 10 Sensors, One Tree........................................................................114
Figure 7.9: 10 Sensors, Two Trees.......................................................................115
Figure 7.10: 10 Sensors, Three Trees.................................................................115
Figure 7.11: 40 Sensors, Two Trees.....................................................................116
Figure 7.12: 40 Sensors, Three Trees..................................................................116
Figure 7.13: 40 Sensors, Four trees.....................................................................117
Figure 7.14: (a) Lifetime Model (Red), Lifetime (Blue), 10 – 70 Sensors; (b) Lifetime Model (Red), Lifetime (Blue), 80 – 130 Sensors.........................................................125
Figure 7.15: (a) R/L Model (Red), R/L Data Points (Blue), 10 – 120 Sensors; (b) R/L Model (Red), R/L Data Points (Blue), 220 – 300 Sensors.......................................130
LIST OF TABLES

Table 2.1: Comparison of Simulation Tools for WSNs.................................22

Table 3.1: Classification and Comparisons of Routing Protocols in WSNs........48

Table 7.1: Number of Sensors, Reliability, Lifetime for NC..........................103
Table 7.2: Mean Number of Trees..................................................................109
Table 7.3: Tree Distributions for 10 Sensors...............................................112
Table 7.4: Two Tree Distributions for 40 Sensors..........................................113
Table 7.5: Expected Distances and Weights, Variable Sensors......................119
Table 7.6: Evaluation of the Proportionality Measure.....................................123
Table 7.7: Evaluation of Reliability per Unit Lifetime.....................................128
Table 7.8: Average Number of Trees and Total Sensors on Trees, Limited Radius...132
Table 7.9: Expected Distances and Weights, Variable Radius.........................133
Chapter 1: Introduction

1.1. Introduction

A wireless sensor network consists of a number of sensors deployed either randomly or in a pre-determined state in a given space. Such sensors are designed to measure one or more physical quantities in the space, such as location or temperature. The sensors need to transmit this collected data to the manager or end-user via internet. Since the sensors concerned are wireless they are typically powered by a battery with a finite lifetime and power output; it may be impossible to recharge or replace such batteries. Thus in a real wireless sensor network a number of parameters naturally need to be considered such as energy consumption, network lifetime. The sensor should be assigned a routing protocol, so that rather than the sensors transmitting directly to the end-user (single-hop protocol), they instead transmit their data via a number of other sensors with the data eventually arriving at the end-user (multi-hop protocol). This thesis seeks to establish relationships between the evaluation parameters (will be detailed in Chapter 4) to show that varying a given parameter does or does not significantly affect another.

Wireless sensor networks are typically used to monitor the environment. The main problems in this primary research area are conserving sensor energy and improving data accuracy.

This thesis will obtain several useful results via simulations. Then, the evaluation models based on different routing protocols in wireless sensor networks will be proposed in this thesis. These evaluation models would allow the deployment of the
sensor nodes to be accomplished without simulations to satisfy the demands of the customer.

1.2. Motivation

A Wireless Sensor Network (WSN) consists of spatially distributed autonomous sensors which co-operatively monitor physical or environmental conditions, such as temperature, sound, pressure, or motion. WSNs are now used in many diverse industrial, military and civilian application areas.

One of the evaluation parameters used to measure the performance of a wireless sensor network is network lifetime (Definition in Section 4.1.2). Due to the remote operation and potential scale of a WSN, replenishing node power (if possible at all) is a time consuming and costly process. One of the approaches to solve this problem is to use natural solar energy from the environment [1]; however, this technique may lead to unreliable node activity (The reason for this can be explained as follows: The power output from the natural sources is highly nonlinear in nature and depends upon a variety of factors, which will lead to unreliable node activity. Although it depends on a variety of factors, service can be provided.). Another useful approach is to add additional redundant nodes that remain in a low power, dormant state until the moment they are required. The trend in reducing both node size and cost means that the environmental and budget impact of the extra nodes can be sufficiently small to permit such an over-deployment. In order to save energy critical nodes can be hibernated, but this may unfortunately disconnect a large region of the network from the base station, resulting in a blind spot.
Mathematical simulation [2] can be used as a tool to help to prolong network lifetime in the field. Results from such simulations are used for modelling the deployment of sensors in a real area. Three questions thus ought to be posed:

(a) Is simulation the only approach to predict the results?
(b) Does a simulation have to be carried out before the deployment of sensor nodes each time?
(c) Is there a better way to make predictions?

Many potential wireless sensor network applications need a balance among energy consumption (E), density (D), radius (Ra), and reliability (R). Trade-offs may indeed exist among above parameters, it would thus be important to develop some mathematical models among these key evaluation parameters. This thesis seeks to develop some evaluation models which will serve to inform the organization of a high efficiency wireless sensor network without prior use of simulations. Such mathematical models will avoid the necessity to undertake simulations for the deployment of sensor nodes resulting in a saving of both time and money.

1.3. Objectives

As mentioned in Section 1.2, equation models will allow users to avoid the necessity to undertake simulations (simulations will waste both time and money) for deployment of sensor nodes. This thesis aims to construct some evaluation models to analyze related parameters.

The parameters (energy consumption, density, radius and reliability) will be analysed in this thesis, and the equation models are designed to show the relationships among these parameters.
As there exist several routing protocols, selecting an appropriate protocol for the application is a problem to be solved. The routing protocols can be divided into three main types: flat protocols, location-based protocols and hierarchical protocols. Single-hop (in Chapter 5), LEACH (in Chapter 6) and Nearest Closer (in Chapter 7) are typical and basic routing protocols for these three types respectively.

This thesis will integrate the Single-hop, LEACH and Nearest Closer protocols into the simulation tool J-Sim and provide a mathematical model for each protocol.

The evaluation models will be constructed then. Thus, the users can select a suitable routing protocol based on the equation models without simulations.

1.4. Contributions of the Thesis

This thesis makes the following contributions:

(a) Some key parameters, such as Reliability and Lifetime, are discussed in the context of different routing protocols.

(b) Comprehensive simulation results have been collected and analyzed based on Lifetime, Reliability, Radius and Density for the Single-hop, LEACH and Nearest Closer routing protocols.

(c) Three detailed evaluation models have been formulated for each of the Single-hop, LEACH, and Nearest Closer routing protocols based on the results of the simulations. In particular, in the LEACH case the models have been demonstrated to be accurate predictors of Lifetime and Reliability, closely matching the values obtained from the simulations.
1.5. Thesis Outline

To achieve the objectives outlined above, the remainder of this thesis has been organized as follows:

Chapter 2 discusses key terms necessary for understanding the aims of this thesis: wireless sensor networks, J-Sim, evaluation parameters and routing protocols. This chapter also reviews literature.

Chapter 3 introduces examples of the three main types of routing protocols, namely flat, hierarchical and location-based.

Chapter 4 defines the evaluation parameters to be used in Chapters 5, 6 and 7 as well as the experimental set-up to be used in those chapters.

Chapters 5, 6 and 7 respectively implement the Single-hop, LEACH and Nearest Closer protocols based on the J-Sim simulation tool and consequently find some useful trade-off analysis results among several key parameters. Based on these results a mathematical model will be constructed for each of the three protocols.

Chapter 8 reviews the thesis and provides some details relating to future work.

Chapter 9 summarizes the results of this thesis.
Chapter 2: Background

This chapter provides an introduction to wireless sensor networks and their applications. Generally sensor nodes are expensive and are difficult to test in large numbers, so that simulation tools become necessary to model networks. The concept of a wireless sensor network and related background information will be detailed in Section 2.1. Some of the existing WSN simulation tools will be described in Section 2.2 and Section 2.3. Several evaluation parameters will be compared in Section 2.4. This chapter will also discuss the problem of energy management and introduce simulation parameters and routing protocols.

2.1. Wireless Sensor Networks

A sensor [3] is a converter that measures a physical quantity and converts it into a signal that can be read by an electronic instrument. For example, a thermocouple converts temperature to an output voltage, which can be read by a voltmeter. Sensors need to be designed to have a small effect on what is measured; making the sensor smaller often improves this and may introduce other advantages. Technological progress allows more and more sensors to be manufactured on a microscopic scale as micro sensors using Micro-Electro-Mechanical Systems’ (MEMS) technology [4–7]. MEMS are made up of components between 1 to 100 micrometres in size and MEMS’ devices generally range in size from 20 micrometres to a millimetre. They usually consist of a central unit that processes data (the microprocessor) and several components that interact with the outside such as micro sensors. Thus MEMS’ technology has allowed the construction of sensor nodes that are capable of sensing, data processing and communicating.
A wireless sensor network [8–9] (WSN) consists of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, pressure, motion. The following figure illustrates the typical data transmission model of such a network.

![Data Transmission Model of a Wireless Sensor Network](image)

Figure 2.1: The Data Transmission Model of a Wireless Sensor Network.

The sensor node transmits the data to the related sensors, and finally the aggregated data arrives at the gateway sensor node or sink node which is connected to the terminal device through internet or satellite. (The terminal device can be used to collect data from the sink node or transmit data to the sink node. It can also be used to interact with manager. A computer is normally working as the terminal device, but a tablet or a smart phone can be the terminal device as well.) Then the manager (end-user) can do analysis.

Wireless sensor networks are used in a diverse range of application areas including industrial, military and civilian. Some of which are listed below:
(a) Military applications [10]: Sensors are widely used in applications such as surveillance, communication from intractable areas to base stations. Since these are inexpensive and deployed in large numbers, loss of some of these sensors would not affect the purpose for which they were deployed.

(b) Distributed surveillance [11]: Highly mobile sensor networks like the underwater autonomous vehicle Zeus, run by Odyssey Marine Exploration Inc. make it possible to transmit huge amounts of data at low power.

(c) Structure monitoring [12]: Structure monitoring systems detect, localize, and estimate the extent of damage. Civil engineering structures can be tested for soundness using such monitoring.

(d) Pollution and toxic level monitoring [13]: Here sensors collect data from industrial areas and areas where toxic spills occur. This is useful in sensing nuclear, biological and chemical phenomena in the environment and transmitting it to remote stations for analysis.

(e) Sensors for vision [14]: This application involves a collaborative self-organizing sensor network, which has many micro sensors built on a chip and implanted in the eye. This improves the vision of people with no or limited vision.

(f) Smart sensor networks [15]: These networks have a number of independent sensors. Each of the sensors makes a local decision, and these decisions are then combined and weighted based on a specific algorithm from which a global decision is made.

(g) Rainfall and flood monitoring [16]: These networks have water level, wind and temperature sensors and the data is transmitted to a central database for analysis and weather-forecasting.

(h) Other applications: Some applications involve habitat monitoring for determining bio-complexity, and others include resource explorations such as mining and mineral analysis [17]. Health applications [18] involve tracking patients and monitoring drug administration in hospitals. Great commercial
opportunities exist in household electronics and in realizing smart home [19–20] and office environments.

Wireless sensor networks have become a part of people’s lives even though few members of the general public are aware of their existence.

The main application of wireless sensor networks that will be simulated in this thesis is target related application, the goal of which is to collect and transmit the data sent from the target node. One target category is individual objects, which usually have a very small size compared with the large area in which the sensor network is deployed, the targets may emit noise, light or seismic waves etc. Another category is continuous objects, which may be spreading in a very large region in which the sensor network is deployed. Nevertheless target related applications share some common characteristics. The data collected by sensors may be redundant or inconsistent and the sensors may have to collaborate on processing the data.

The following section will introduce several aspects of WSN management that are of particular interest in this thesis.

2.1.1. Media Access Control

A MAC layer is required to assign the communication channel to the sensor nodes as the sensor nodes cannot share the same channel without a data collision occurring. A data collision is the simultaneous presence of signals from two or more sensors on the network. A collision occurs when two or more sensors each think the network is idle and start transmitting at roughly the same time.

A MAC protocol for wireless sensor networks should ideally consume little power, avoid collisions, be implemented with a small code size and memory requirement, be efficient for a single application, and be tolerant to changing radio frequency and
networking conditions. One general contention-based strategy is for a transmitter to test the channel to see if it is busy, if not, it transmits; otherwise it waits and tries again later. If two or more nodes transmit at the same time, which can occur if they are hidden from each other, then there is a data collision and all the transmitted data from them is considered lost (it may be possible to retransmit it at a later time).

A frame is a digital data transmission that includes frame synchronization i.e. a sequence of bits or symbols making it possible for the receiver to detect the beginning and end of the packet in the stream of symbols or bits. A node wishing to send data initiates the process by sending a Request to Send frame (RTS). The destination node replies with a Clear To Send frame (CTS). Any other node receiving the RTS or CTS frame should refrain from sending data for a given time. The amount of time the node should wait before trying to get access to the medium is included in both the RTS and the CTS frame. IEEE 802.11 uses RTS/CTS acknowledgment and handshake packets to partly overcome the hidden node problem, that is, nodes that are out of range of other nodes or a collection of nodes, but that are in range of a cluster head or the sink node. The exposed node problem occurs when a node is prevented from sending packets to other nodes due to a neighbouring transmitter. The IEEE 802.11 RTS/CTS mechanism helps to solve this problem only if the nodes are synchronized and packet sizes and data rates are the same for both the transmitting nodes.

2.1.2. Time-Division Multiple Access

One approach to MAC design for sensor networks is to use Time-Division Multiple Access (TDMA) [21] based protocols that conserve more energy than contention-based protocols like Carrier Sense Multiple Access. TDMA allows several users to share the same frequency channel by dividing the signal into different time-slots. It has a natural advantage of collision-free medium access. The approach of TDMA in a MAC layer is also the basic technique to realize the some hierarchical routing protocols. The MAC layer provides the RTS/CTS mechanism to control
communication among different sensor nodes. Unfortunately these management signals can be dropped due to collisions. These dropped packets may lead to an error for the sensed data. In a single TDMA schedule, the time-frame length is defined to be

\[ \text{schedule size} \times \text{spread-spectrum packet transmission time} \] (1)

The schedule size is essentially just the number of sensors in the cluster at that time and so is a dynamic variable. On the other hand the spread-spectrum packet transmission time is a constant value, which in the experiments conducted with two cluster heads and 20 sensors equalled 0.0078 time slots. Thus the time-frame length varies in direct proportion to the schedule size. The spread-spectrum packet transmission time is defined in turn as

\[ \text{slot time} \times \text{spreading factor} \] (2)

Packet transmission time is the amount of time from the beginning until the end of a packet transmission. Spreading factor means the number of chips that are used to spread one data symbol (The bits in the spreading code are called chips to differentiate them from the bits in the data sequence, which are called symbols). Here the slot time is the packet transmission time and the spreading factor in J-Sim is 1 plus the integer part of \((1.5 \times \text{number of clusters})\). The main use of the spreading factor is to take into account the number of clusters by making the time-frames long enough to avoid data collision between transmissions from different cluster heads. The figure of 1.5 times the slot time may be interpreted as allowing one slot time per sensor in the cluster to allow for collection of transmitted data from each such sensor, and then a further half a slot time per sensor in the cluster for all the packets of data received by the cluster head in that time-frame to be aggregated, compressed and transmitted to the sink node. In reality, the slightly longer than one time slot per cluster sensor will
be allowed for data collection to take into account that a sensor cannot transmit its data at an exactly prescribed time. Thus the compression factor is at worst 50%. Adding the one in the formula for the spreading factor ensures that there is adequate time for these processes to occur, particularly when the number of clusters is small and odd, but clearly becomes less significant as the number of clusters increases.

2.1.3. Slotted ALOHA

Slotted ALOHA [22–25] is a type of TDMA transmission system developed at the University of Hawaii in the early 1970s. Slotted ALOHA improves contention management through the use of beaconing, in which each sensor transmits at precise synchronized intervals, indicating to its source sensors within the transmission radius when the channel is clear to send a packet of data to it. The time axis is divided into slots of duration equal to the transmission time of a (fixed length) packet plus the guard time. Every sensor starts transmission of its packet at the beginning of the slot it chooses, so a packet arriving to be transmitted at any sensor must delay until the beginning of the next slot and in particular some slots will remain idle. At any time, each sensor may either transmit or receive, but not both. Every packet transmitted contains the address of the receiving sensor and a sensor will discard packets that are not addressed to it. Two or more transmissions to the same receiving sensor in the same slot will result in a collision and all such transmissions are considered failed. Slotted ALOHA has the advantages of being simple and that a single active sensor can nearly continuously transmit at the full channel rate.

2.2. Wireless Sensor Network Simulation Tools

Wireless sensor networks are composed of many sensor nodes. Owing to the limited number of sensors that can be distributed in a real experimental network means that much research in this area is based on an over-simplified analysis, and therefore only
limited confidence can be placed on predictions arising from such experiments. Thus simulation has become a common way to test new applications and new protocols before real deployment [26]. Simulation results rely not only on the environment but also on the physical layer assumptions, which are not that accurate. Although this problem exists, simulation is still a good approach for the deployment of real sensors.

This thesis will do simulations for different routing protocols, so the ability to obtain reliable results (such as network lifetime, number of packets transmitting via sensors, etc.) is one of the features the simulation tool should have. There is no need for the simulation tool to automatically generate useful topology diagrams. Some evaluation models are constructed based on the results of network Lifetime, so the power consumption model is necessary in the simulation tool. Typical and basic routing protocols are widely used. Based on the simulation results of these routing protocols, several useful evaluation models can be constructed and these routing protocols should be implemented in this work. So an open-source simulation tool is important. At this stage, there is no need to support particular routing protocols. The simulation tool should be a discrete-event simulator and therefore can run for as long as necessary to complete the simulation. The scalability is an important factor for the users to choose the simulation tool. In this thesis, 300 nodes application should be simulated. Based on the simulation results, mathematical model can be constructed. Thus, the simulation tool should support at least 300 nodes simulation. The sensors in the experiments of this thesis are static, and are randomly placed. Though not a strict requirement, it would be an advantage for the simulation tool to support automatic random placement.

Future work of this thesis aims to integrate the power consumption model into SunSpot sensors, so a Java-based simulation tool should be selected.
The following is a general model of a simulation tool. The model includes several sensor nodes, a radio channel, environment, agents and sink nodes.

![Simulation Model Diagram](image)

Figure 2.2: Simulation Model [27].

A detailed description of the components in this figure is as follows:

(a) Nodes: Nodes are basic devices in this model. Each node can communicate with each other through the radio channel. There also exists a protocol stack to control these communications.

(b) Environment: The environment component models the generation and propagation of events that are sensed by the sensor nodes, which can lead to other sensor actions.
(c) Radio channel: This component characterizes the propagation of radio signals among the nodes in the network.

(d) Sink nodes: The sink nodes will receive data from common sensor nodes.

(e) Agents: Agents play a role as a generator of events of interest for the nodes.

Based on this simulation model, several examples of popular simulation tools will be analyzed from Section 2.2.1 to Section 2.2.7.

2.2.1. NS-2

*Network simulator version 2* [28–29] (NS-2) is a discrete event simulator targeted at networking research, which is developed in C++. C++ is one of the most popular general-purpose programming languages and is implemented on a wide variety of hardware and operating system platforms.

NS-2 supports the *Transmission Control Protocol* (TCP). TCP [30–31] is one of the original core protocols of the internet protocol suite and resides at the transport layer; it provides routing and multicast protocols over wired and wireless networks. OTcl [32–34], an object-oriented dialect of Tcl has been used as configuration and script interface in NS-2. Tcl [35–37] is a scripting language, which is a programming language that supports the writing of scripts; these are programs written for a special runtime environment that can interpret and automate the execution of tasks.

NS-2 can implement protocols such as Directed Diffusion or *sensor medium access control* (S-MAC). S-MAC is a MAC protocol specifically designed for wireless sensor networks, which uses less energy than the standard protocol. In addition, there exist some projects like *SensorSim*, which plan to provide wireless sensor support to NS-2. One of NS-2’s disadvantages is that it is based on a flat-earth model in which it assumes that the environment is flat without any bulge or sinking. Also,
NS-2 graphical support is not very good and the scalability of NS-2 is limited. This thesis will select a Java-based simulation tool, so NS-2 is not the choice.

2.2.2. OMNET++

OMNET++ [38–42] is an extensible, modular discrete event simulator written in the C++ language. The framework of OMNET++ provides a very strong functionality as it supports queuing networks, performance evaluation and communication networks. It also provides support for real-time simulation, network emulation, database integration and many other functions. Compared to NS-2, OMNET++ provides a better graphical user interface (GUI) library for tracing and debugging support. In addition the MAC protocol has been implemented in OMNET++ and it can simulate power consumption.

![OMNET++ Interface](image)

**Figure 2.3: OMNET++ Interface [43]**

The disadvantage of this simulation tool is that it doesn’t provide many protocols in its library. However, OMNET++ has become a popular simulation tool for wireless
sensor networks. This thesis will select a Java-based simulation tool, so OMNET++ will not be the choice.

2.2.3. TOSSIM

TOSSIM [44–45] is a discrete event simulator for TinyOS sensor networks. Users can compile TinyOS code (nesC) into the TOSSIM framework, which runs on a PC. Thus, users can debug and test different algorithms in a repeatable and controlled environment.

TinyViz is the user-interface of TOSSIM, which provides users with a convenient and highly efficient running environment. This means that there is no need for a user to go through all the commands. The following figure illustrates the TinyViz interface.

![TinyViz Interface](image)

Figure 2.4: TinyViz Interface [46].
TOSSIM is designed to simulate TinyOS networks instead of the real world. In other words, TOSSIM will just focus on the behaviour of TinyOS. The TOSSIM framework can simulate a huge number of sensors; in addition TOSSIM has a good radio model. Thus many simulation results can be obtained by using TOSSIM. On the other hand, TOSSIM cannot provide users with a good energy consumption simulation, which is the main disadvantage of this simulator. So this thesis will not choose TOSSIM as its simulation tool. A second disadvantage is that the programming language NesC (the dialect of C previously mentioned) should be run on all the sensor nodes, so TOSSIM can only simulate homogeneous applications.

2.2.4. ATEMU

ATEMU [47] is a C-based simulation tool for wireless sensor networks, which operates under Linux. The processor of ATEMU is called AVR, which is used in MICA2 sensors. The GUI for ATEMU is named XATDB, which provides a good interface for users to know the actions of sensor nodes. Figure 2.5 shows an example of a six sensor node simulation. ATEMU is designed to bridge the gap between real sensors and simulation, as it can not only be implemented for real sensors but also provides users with a simulation for the interactions among the sensor nodes.
The main advantage of ATEMU is that it supports a heterogeneous sensor network. That is to say it is possible not only to simulate on the MICA2 nodes, but also on other platforms. Based on the ATEMU simulation a lot of useful and accurate results can be obtained, which assists users to find unbiased comparisons. The main disadvantage of ATEMU is that it only supports a limited number of routing protocols, so, for example, it doesn’t provide any routings regarding the clustering problem. This thesis will not select a real-time only simulation tool as a discrete-event simulator can run for as long as necessary to complete the simulation. Thus ATEMU is not the choice of simulation tool in this thesis.

2.2.5. EmStar

*EmStar* [48] is a C-based simulation tool for wireless sensor networks, which operates under Linux. EmStar provides the users with a GUI (Figure 2.6) with which users control the devices directly.
The main advantage for EmStar is that the debug operation for EmStar is very convenient and users can switch the simulation and deployment of sensors freely. The main disadvantage of EmStar is that it can only run in real-time. However, a discrete-event simulator can run for as long as necessary in order to complete the simulation. So EmStar will not be selected as the simulation tool.

![EmStar Interface](image)

Figure 2.6: EmStar Interface [49]

Figure 2.6 is a screen shot of EmView, the Emstar visualizer.

### 2.2.6. J-Sim

**J-Sim** [50] (formerly known as JavaSim) is an open-source, component-based compositional network simulation environment. The system is based on the IEEE 802.11 [51] implementation provided with J-Sim. IEEE 802.11 is the first wireless LAN (WLAN) standard proposed in 1997. J-Sim provides a script interface that allows its integration with Tcl and has been developed entirely in Java. Java is a
general purpose object-oriented computing language that is specifically designed to have as few implementation dependencies as possible.

The main advantage of J-Sim is that it provides some basic routing protocols, such as Greedy Perimeter Stateless Routing [52] and Directed Diffusion [53], and it also provides the user with a wireless sensor network simulation framework with a very detailed model of a WSN.

In the J-Sim framework a five-layer sensor stack and a power model are basic components for the key component, the sensor node. A diagram explaining the structure of the sensor node is:

![Diagram of Sensor Node](image)

Figure 2.7: Sensor Node [50].
2.2.7. Comparison of Simulation Tools

The following table compares some popular simulation tools [54] (Include: NS-2, OMNET++, TOSSIM, ATEMU, EmStar, and J-Sim) and illustrates their advantages and disadvantages.

Table 2.1: Comparison of Simulation Tools for WSNs.

<table>
<thead>
<tr>
<th>Tool</th>
<th>GUI</th>
<th>OS Type</th>
<th>Open-source and online documents</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS-2</td>
<td>No</td>
<td>General</td>
<td>Yes</td>
<td>Several routing protocols have been implemented. It supports randomly placement of sensors</td>
<td>Graphical support is not very good and the scalability is limited.</td>
</tr>
<tr>
<td>OMNET++</td>
<td>Yes</td>
<td>General</td>
<td>Non-commercial or commercial license</td>
<td>MAC protocol has been implemented. It can simulate power consumptions. It supports randomly placement of sensors</td>
<td>It doesn’t provide many protocols in its library.</td>
</tr>
<tr>
<td>TOSSIM</td>
<td>Yes</td>
<td>Specific</td>
<td>Yes</td>
<td>The scalability is good.</td>
<td>It can only simulate homogeneous networks.</td>
</tr>
<tr>
<td>ATEMU</td>
<td>Yes</td>
<td>Specific</td>
<td>Yes</td>
<td>It can support heterogeneous networks. The power consumption model is good.</td>
<td>The simulation time is long.</td>
</tr>
<tr>
<td>EmStar</td>
<td>Yes</td>
<td>Specific</td>
<td>Yes</td>
<td>The debug operation is very convenient.</td>
<td>It can only run in real-time.</td>
</tr>
<tr>
<td>J-Sim</td>
<td>Yes</td>
<td>General</td>
<td>Yes</td>
<td>It provides some routing protocols. It has a good GUI. The power consumption model has been implemented. The scalability is good. It is a Java-based simulation tool. It supports randomly placement of sensors</td>
<td>The simulation time is long.</td>
</tr>
</tbody>
</table>
This thesis will select J-Sim as its simulation platform and the details of J-Sim will be further analyzed in Section 2.3.

2.3. Details of J-Sim

2.3.1. J-Sim Platform

J-Sim was selected as the simulation tool of choice for the following reasons:

(a) The authors of J-Sim have performed detailed performance comparisons [55] in simulating several typical WSN scenarios in J-Sim and NS-2. The simulation results indicate J-Sim and NS-2 incur comparable execution time, but the memory allocated to carry out simulation in J-Sim is at least two orders of magnitude lower than that in NS-2. As a result, while NS-2 often suffers from out-of-memory exceptions and was unable to carry out large-scale WSN simulations, the proposed WSN framework in J-Sim exhibits good scalability.

(b) J-Sim models are easily reusable, so users can combine the components in the framework freely. J-Sim also provides a GUI, which makes it easy to operate the simulation.

J-Sim is a Java-based platform. The Java-based sensors could be integrated with Java-based simulation tools in the future.

2.3.2. Energy Consumption Models

There are a number of energy consumption models used in simulations, this work will detail the one that J-Sim [50] uses and then discuss some other models.
In J-Sim the energy consumed in a single hop in transmitting a packet of data from one sensor to another is considered to be proportional to the second power of the wireless transmission distance. Thus the expected energy consumption of one sensor in transmitting to the sink node is given by

\[ O(E(h)E(d^2)) \]  

(1)

where \( E(h) \) is the expected number of hops from a sensor to the destination and \( E(d^2) \) is the expected value of the square of the distance between sensors. This of course means that short-hops consume less energy than long ones.

An alternative, but indirect measure, is called the *log-distance model* [56]. In this model the power received by a node a distance of \( d \) from the sender can be expressed as:

\[ P_0 \times (d_0/d)^{\alpha} \]  

(2)

where \( P_0 \) is the power of the signal received at distance \( d_0 \) from the sender and \( \alpha \) is the so-called path-loss exponent. The value of \( \alpha \) depends on the specific propagation environment and experiments have shown that usually \( 2 \leq \alpha \leq 5 \). In this model nodes communicate with the minimum power necessary to reach the destination, with the requirement that the signal at the receiver is above some set sensitivity level. The power consumed is then assumed to be a linear function of the power received given by the formula above.

It would appear in both of these models that short-hop routing should lead to reduced energy consumption and higher signal to interference ratios (SIR). (Here SIR is the signal strength received by a device divided by the interfering strength of other simultaneous transmissions).
If a long-hop of distance $d$ is divided into $h$ hops of distance $d/h$ [57], the energy benefit is assumed to be $h^{\alpha-1}$, where $\alpha$ is the path-loss exponent. However, this assertion is often based on a crude channel model that ignores delay, end to end reliability, bias power consumption, the impact of channel coding, mobility and routing overhead. This work concludes that in many cases long-hop routing can rival short-hop in reality.

2.3.3. Reliability Models

The simplest model for reliability within wireless networks is the *disk model* [58]. Here the transmission of a packet from one sensor to the next in the protocol is either a complete success or a complete failure, depending on whether the distance is larger or smaller than the transmission radius and whether a data collision doesn’t or does occur respectively. This is the model that J-Sim adopts. It has the advantage of simplicity making it easy to model, but it has the disadvantage of being unrealistic in complicated real-world applications (The reason why it’s unrealistic in real world applications can be explained as follows: In the real world, the state of sensors may be changed by some unpredictable reasons, so the process of transmission of a packet from one sensor to the next in the real-world applications may not the same as that in the model. However, this is a drawback of all simulation tools).

The second, and more sophisticated, model is the *threshold model*, such as the Rayleigh fading model [59–60]. Here a certain *signal-to-interference-and-noise ratio* (SINR) (SINR means the signal strength received by a device divided by the interfering strength of other simultaneous transmissions (plus the fixed background noise $N$)) is needed for successful transmission.

It is unclear whether more interference is caused by a single transmission at higher power (long-hop) or multiple transmissions at lower power (short-hops). The routing schemes with the least energy consumption will also be those causing the least
interference [61]. It is then a matter of designing a MAC scheme to take advantage of this reduced interference.

If all the nodes increase their power by the same factor, then the SIR levels remain constant but the SINR levels increase. Thus this does not have a negative impact on packet reception in the network using the threshold model. In contrast in the disk model it would be predicted that more collisions would occur, because the transmission radius (the distance between the sensor and furthest point to which the sensor can transmit the information) will grow with the transmission power and any interferer within a receiver’s disk is assumed to cause a collision.

In the threshold model dividing the distance into \( h \) hops increases the SNR (the signal power divided by the noise power) at each hop by \( h^\alpha \), where \( \alpha \) is the path-loss exponent; however, the end-to-end rate is decreased by a factor of \( h \). So the \( h \)-hop routing protocol needs to transmit at an \( h \) times higher per hop rate to achieve the same end-to-end throughput. The rate loss is linear, whilst the increased SNR is proportional to \( \log h \), so there exists an optimim \( h \) for each end-to-end rate. So for example in low power transceivers the local oscillators and bias circuitry will dominate energy consumption, so that short-hop routing does not yield any substantial energy benefit [62]. Clearly the path efficiency: the ratio of the Euclidean distance between the end nodes and the actual transmission distance travelled will be higher if long hops are used. The variance of hop length leads to increased variability in energy consumption. In fact path efficiency increases monotonically with \( m \) if only every \( m \)th nearest node is used as a relay and goes to 1 for large \( m \) [63], whereas it is \( 1/\sqrt{2} \) for nearest neighbour routing [64]. The lifetime of an \( h \)-hop route is considered in [65], it is determined there by the node that consumes the most energy and this grows at a rate equal to at least \( \log h \).
2.4. Evaluation Parameters

Many researchers have performed simulations for target related applications in WSNs given their significant role in the efficiency of military and civil applications such as environment monitoring and target surveillance [66–70]. These researchers have analyzed the relationship among some evaluation parameters, but most results in their published work are to minimize energy consumption or analyze the relationship between two parameters. No mathematical model with three or more parameters has been proposed so far. Some evaluation parameters will be defined in Section 4.1 and some mathematical models will be proposed in Chapter 5, 6 and 7.

The following sub-sections will detail some of the published work on the evaluation parameters.

2.4.1. Energy-efficiency Problem

Within a WSN, sensors send packets of data to other sensors or the sink node. A packet is a formatted unit of data; the data consists of control information and user data or payload. The control information provides data the network needs to deliver the user data, such as source and destination addresses and checksums, and is typically found in packet headers and trailers with payload data in-between. As the names suggest header and trailer data can be found at the beginning and end respectively of the packet. The payload data is the fundamental purpose of the transmission to the exclusion of information sent with it.

The principle of prediction-based dynamic energy management includes three main aspects:
(a) **Dynamic Awakening Mechanism**: This dynamic awakening mechanism adopts the approach of an awakening state and takes node idle time into account as well. The Particle Filters (PF) algorithm predicts the target state. Each node then uses the predicted target state to estimate idle time so that it can keep the idle period as long as possible to save energy.

(b) **Distributed Genetic Algorithm (GA) and Simulated Annealing (SA)**: GA encodes the parameters into finite bit strings. Each of these strings provides a possible solution to the problem and then works with a set of strings. SA starts with a random solution state and then generates further states iteratively from it.

(c) **Forwarding Node**: In wireless sensor networks the sensor nodes next to the target can acquire information and transmit it to a sink node in each sensing period.

Wang et al [71] focused on the energy-efficiency problem in WSNs, they not only proposed a dynamic energy management mechanism based on target prediction, but also built a collaborative sensing and energy consumption model. This can predict the target location using the Particle Filters approach; based on this location information the sensor nodes can update their timetable and sleep without losing any event. Meanwhile, as the candidate nodes for sensing are known beforehand, the researchers accomplished the optimization of the sensing process using a hybrid of GA and SA, which uses the distributed computing capability of WSNs so that energy consumption can be minimized without degrading the performance accuracy. Moreover, a routing scheme of forwarding nodes was suggested to achieve extra energy conservation. This literature, together with many others, has introduced lots of approaches to conserve energy, but it would not be very easy to apply all of these approaches in a real system. (There are several limitation factors, such as, the space limitation for the deployment of sensors, the effect of uncertain environment for the network, etc.) After performing the simulations the researchers above compared the energy consumption from these
different approaches. Although these policies are based on not degrading the performance accuracy, in their research no analysis between energy consumption and accuracy (Sensing accuracy is defined by the error ellipse.) was performed.

In [72], the researchers propose a novel uneven clustering scheme (DEUC) to construct the wireless sensor network. From the analysis of simulation results, the authors conclude that DEUC can effectively balance the energy consumption of cluster heads and extend the network lifetime.

2.4.2. Radius and Number of Nodes

In [2], Maity & Gupta considered a randomly distributed wireless sensor network covering a large area. They wished to find an estimate for the number of nodes required with the minimum critical communication distance to ensure network connectivity and stability. Using results from graph theory certain mathematical formula-based algorithms already existed for a relatively small number of sensors; however, the authors proposed a new formula based on mathematical simulation, which minimized the inter-node critical radius prediction for a large number of nodes. They chose MATLAB as their simulation tool and constructed a regression equation between the radius and number of nodes. The theoretical model provided better results if the number of nodes is less than 250, but their regression equation provided smaller answers for the radius to preserve connectivity for larger numbers of nodes.

2.4.3. Coverage and Lifetime

Since sensors may be spread in an arbitrary manner, one of the fundamental issues in a wireless sensor network is the coverage problem. In general, this reflects how well an area is monitored or tracked by sensors. As pointed out in [73], the coverage concept is a measure of the quality of service (QoS) of the sensing function and is subject to a wide range of interpretations due to a large variety of sensors and applications. The aim in this work is to make each location in the physical space
within the sensing range have at least one sensor node. The goal for the authors to address the target coverage problem in [74] is to prolong the network lifetime of an energy-constrained wireless sensor network. The sensor nodes are deployed randomly around the target and if the target enters into the sensor’s sensing range, then the sensor will get the target location information and send it to a central processing node.

In order to prolong the lifetime of the network the authors divide the sensor nodes into several sets in which all the targets can be covered by the sensors in each set. To conserve energy and extend the lifetime of the sensor network these sensor sets are activated successively, such that at any time only one set is active (the active state contains transmit, receive and idle). The nodes in the other sets will be in a low-energy sleep state. Sensor nodes can adjust their state between active and sleep, which will extend network lifetime compared with the case when all sensors are active. At the same time the number of active sensors in the application area will decrease, which will result in reducing contention at the MAC layer.

In this thesis, generally the sensors will, by design of the transmission radius, be able to cover the network geographically and so the coverage problem, in this instance, might be interpreted as the length of time for which this coverage is maintained. It should be noted that the experiments on network lifetime in this thesis do not directly address this problem; however, some (limited) experiments will consider the situation when the transmission radius of a sensor is insufficient to cover the whole network and measure the reliability of the resulting system.

2.4.4. Energy-efficient Protocol

In the literature [75], the researchers propose an energy-efficient protocol which includes two algorithms: RARE-Area (Reduced Area Reporting) to limit the sensors according to the quality of data and RARE-Node (Reduction of Active Node
Redundancy) to reduce the redundant information to be sent to the sink node. The operation of the protocols is as follows:

When the sensor detects a target, information can be generated by the target and then the weight will be calculated by the Reduced Area Reporting algorithm. If the weight is greater than or equal to the upper threshold, which was set at the beginning of tracking, the sensor node will send a beacon message to all related sensors and wait to receive at least two beacons from other sensors that have detected the target as well. Using this information the sensor can calculate the target position.

The following two cases need to be considered:

(a) If only the RARE-Area algorithm is implemented, once the target position has been calculated the sensed information will be forwarded to the cluster head.

(b) If both the RARE-Area and RARE-Node algorithms are implemented, they proceed to check for data redundancy. In this case they need to check if there exists any extra sensor node within the sensing range. If not, the information will be transmitted to the cluster head directly as it is not redundant information. In the other case, they should confirm whether the information sensed by the current sensor is redundant or not, if not they will transmit the information to the cluster head.

The simulation results compare the lifetime with each of the algorithms. The simulation studies showed that with medium values of the threshold weight, the implementation of the RARE-Area protocol alone or using both the RARE-Node and RARE-Area algorithms together decreased energy consumption without decreasing tracking accuracy more than 20%.
For the simulation parameters the researchers compared the relationship between the number of dead nodes and network lifetime and in addition they plotted the data accuracy value over time.

In [76], the researchers propose a centralized routing protocol called base-station controlled dynamic clustering protocol (BCDCP), which distributes the energy dissipation evenly among all sensor nodes in order to improve network lifetime and decrease average energy consumption. The researchers also compare BCDCP to the cluster-based routing protocol called LEACH. The simulation results show that BCDCP can reduce the average energy consumption and improve network lifetime.

The literature [77] introduces an energy-efficient routing protocol based upon clusters using message success rate. From the analysis, the researchers conclude that this protocol can provide better energy efficiency than existing approaches.

2.4.5. Energy-quality Trade-offs

In [78], the researchers introduce four different activation approaches to switch on some sensors in the wireless sensor network: naive activation, randomized activation, selective activation based on trajectory prediction and duty-cycled activation. Using the above approaches they showed that energy savings come at the expense of a reduction in the quality of tracking, in particular increasing the sensor range leads to a decrease in target tracking accuracy. The four activation approaches are:

(a) *Naive activation*: In naive activation all nodes in the network are in tracking mode all the time. While clearly this strategy offers the worst energy efficiency it is a useful baseline for comparison, because it provides the best possible quality of tracking.
(b) *Randomized activation*: In this strategy each node is on with a probability $p$, so that on average a fraction $p$ of all the nodes will be on and in tracking mode.

(c) *Selective activation based on prediction*: In this strategy only a small subset of all the nodes contains nodes in tracking mode at any given point of time. These nodes also predict the next position of the target and hand over tracking to nodes best placed to track the target next. The rest of the nodes are in communication mode and can switch to tracking mode on being alerted by signals from the tracking nodes. The basic idea of selective activation based on prediction is to use the previous history of the location of the target (as predicted by the sensors) to determine its next predicted position.

(d) *Duty-cycled activation*: In this strategy the sensor nodes in the network will be switched on and off periodically; this can be applied together with any other activation strategy.

In fact this work showed that using selective activation is the best strategy, since it provides significantly reduced traffic error for low energy expenditure.

In [79], a centralized tracking filter and procedure for selectively activating sensors around the expected target’s position are combined. Unlike selective activation methods existing in [79], which are concerned only with tracking accuracy, [80] seeks to trade-off tracking performance optimization against lifetime maximization.

2.4.6. **Extend Network Lifetime**

In the literature [80], the implementation for the system is to increase sensor network lifetime. The goal of the system is to improve the ability to prolong the lifetime of the wireless sensor network and to track the targets using a low energy consumption strategy.
The system uses sleep mode and active mode for each sensor to make them collectively conserve energy. Many different sensors (photo-detecting sensors, acoustic sensors) were used in this system. The two processing modes for sensors are active and sleep.

The acoustic sensors are in sleep mode at first and were adjusted into active mode if the target occurs. Using this strategy energy was conserved and the lifetime of the whole network was thus extended. The researchers divided the whole sensor network into different regions, so the approach the authors applied is decentralized target tracking, with one cluster head in each region. A cluster head can collect the sensed information from all active sensors in the cluster. After obtaining this information the cluster head will process and transmit it to the base station. The normal nodes in each region are not able to process this sensed information, they can only transmit this information to the cluster head.

A base station is a sensor that can transmit data through the internet and hence worldwide. In addition to transmitting information the base station node can also connect with the cluster heads. The base station in the author’s system can predict the next target location and transfer the location data to the end-user.

In this system not all the sensors in the wireless sensor network need to be in an active mode. The nodes in the area where a target arrives certainly need to be in active mode while the others will go into sleep mode in order to conserve energy, since sensor nodes in active mode use much more energy than those in sleep mode.

The dynamic clustering approach provides lots of benefits, but is based on minimizing the energy consumption of sensors. These sensors need to discover the next cluster head and form the next cluster. The users may choose different clustering approaches
(static clustering approach or distributed approach) depending on the requirements of tracking quality and energy consumption.

The simulation results show us the value of energy consumption as the density of sensors increases. The researchers also compared the effect of different clustering approaches on the above parameters. If energy can be conserved, obviously lifetime can in turn be prolonged.

Lifetime is a significant parameter to evaluate a sensor network. This thesis will compare some different routing protocols. One of these protocols is LEACH, which is a cluster-based protocol. In the literature [80], the simulation provides us with the relationship between the number of nodes and energy consumption, but the researchers haven’t done any further research for any other evaluation parameters.

In [81], the researchers have detailed the following approaches to extend network lifetime:

- (a) mobile sinks: sensors are set close to the sink and then some shorter multi-hop data transmission paths can be obtained to save energy consumption;
- (b) mobile sensors redeployment: the sensor nodes can be redeployed to save energy and extend network lifetime;
- (c) mobile relays: the mobile relays can carry data to the sink in order to save transmission energy consumption.

All the above approaches are based on the nodes’ mobility technology.
In [82], the researchers propose approaches to adjust the transmission range of sensors and relocate the sink node. The analysis show that these approaches can prolong the network lifetime of the WSN greatly.

In the literature [83], the authors show that organizing sensors in cooperative groups can save energy of the wireless sensor network. It can also reduce the number of the messages transmitted inside the sensor network. The simulation results will show how the number of groups extends the network lifetime.

2.4.7. Energy, Density, Latency, Accuracy Trade-offs
In the literature [84], the researchers concluded that different parameters could be ‘tweaked’ in order to achieve better accuracy or longevity. The relationships between these parameters are not limited to tracking applications. The researchers examined the localization accuracy of a wireless sensor network with different data latency, target speed and application deadline. In order to get good accuracy the routing protocol needed to be selected. In turn with a suitable choice of routing protocol in place, the tracking application parameters were adjusted to obtain better accuracy. In the simulations the researchers showed the relationship between accuracy, radius, target speed and lifetime of the application.

2.4.8. Algorithms for Better Accuracy
In the literature [85], the researchers introduce a new algorithm called Optimized Communication and Organization (OCO). This algorithm ensures very good accuracy of target tracking and efficient energy dissipation for sensor nodes.

OCO has four phases: position-colllecting, processing, tracking, and maintenance:

(a) Position-colllecting: In this phase the base nodes collect the position information for all the sensor nodes.
(b) **Processing:** In this phase, the base node cleans up the redundant nodes, detects the border nodes (Nodes that are positioned along the border of the network area are called border nodes [86]) and routing for transmitting data.

(c) **Tracking:** In this phase all objects are detected that are near the perimeter of the sensor network. The border nodes are switched on. If one of the border nodes detects an object, it periodically sends its position information to the base node. However, if this object is lost, the border node will send a message to all its neighbours to turn them on. If a neighbour detects the object, it will continue to send the position information to the base node. On the other hand, if no object can be detected by the neighbour nodes, this packet transmission process will be terminated and all the neighbour nodes sensor modules will be switched off.

(d) **Maintenance:** If the network has a dead node (energy has run out), this phase begins. In this situation the base node will delete this dead node and will arrange to restart this four step process again.

Simulation evaluations of OCO were compared with two other methods under various scenarios. These results show that OCO produces better accuracy results than the other two methods.

In the literature [87], an updated DV-Hop algorithm that aims to improve localization accuracy has been proposed. Compared to the original DV-Hop algorithm, the researchers use corrected value of global average per-hop error and global average per-hop distance to replace the average per-hop distance. From the simulation results the researchers conclude that the updated DV-Hop algorithm can improve the accuracy of sensor nodes localization more effectively.
2.5. Summary

This chapter has introduced some formative descriptions of wireless sensor networks and also provided some details about J-Sim. In addition, this chapter has included a review of work on WSN evaluation parameters. In some of the papers the researchers have performed some simulations and obtained some useful results with related evaluation parameters. However, none of these have constructed a mathematical model for those related parameters nor have the trade-off approaches been discovered. Thus simulation tools will not be necessary for the deployment of sensors resulting in a saving of both money and time.

Furthermore, in several published papers, researchers have proposed some new protocols to improve the lifetime of the network and conserve energy. The subsequent chapter will analyze differing routing protocols.
Chapter 3: Routing Protocols

In this chapter, routing protocols can be divided into flat, hierarchical and location-based routing protocols [88] depending on the network structure. In flat routing protocols all the sensors are assigned the same roles. In hierarchical-based routing protocols different energy level sensors will be assigned different roles. In location-based routing protocols the sensors’ location information will be used to transmit data.

3.1. Flat Routing Protocols

In flat networks it is hard to assign an identifier to each sensor, as all the nodes play the same role and collaborate together to perform the sensing task and the number of sensors is generally large. The following sections will summarize five flat routing protocols and highlight some of their advantages and disadvantages.

3.1.1. Single-hop Protocol

This is a simple protocol in which sensors transmit directly in one hop to the sink node. Latency will thus be minimized, but data collision may occur at the sink node and sensors situated a long way from the sink node may expend a large amount of energy in transmitting to the sink node.

3.1.2. Sensor Protocols for Information via Negotiation (SPIN)

SPIN [89] is a family of protocols that enable a user to query any node and obtain any required information immediately. One of the advantages of SPIN is that topological changes are localized, since each node needs to know only its single-hop neighbours.
SPIN provides much energy saving over a flooding protocol (see below). However, SPIN’s data advertisement mechanism cannot guarantee the delivery of data.

The disadvantages of SPIN are as follows: Firstly, it isn’t scalable; secondly the nodes around a sink could deplete their battery quickly if the sink is interested in too many events. Thirdly, for a given localized event, the data may be sent throughout the entire network.

3.1.3. Directed Diffusion (DD)

DD [90] is a data-centric protocol, that is, the primary function of the protocol is the management of data. Data generated by sensor nodes is named by attribute-value pairs. A node requests data by sending interests for named data. Data matching the interest is then ‘drawn’ down towards that node. Intermediate nodes can cache, or transform data, and may direct interests based on previously cached data. All sensor nodes in a directed diffusion-based network are application-aware, which enables diffusion to achieve energy savings by selecting empirically good paths and by caching and processing data in the network. DD uses flooding to send the query information around the network—in general flooding is a simple routing technique in which a sensor transmits its data to all eligible receivers.

The performance of data aggregation methods, used in the directed diffusion paradigm, are affected by several factors that include the positions of the source nodes in the network, the number of sources, and the communication network topology.

The disadvantages of this protocol can be described by the following two aspects. Firstly, to implement data aggregation it employs a time synchronization technique, which is difficult to realize in a sensor network. Secondly, data aggregation is the overhead involved in recording information. These two points may lead to increasing the cost of a sensor node.
3.1.4. Rumor Routing (RR)

RR [91] is a variation of Directed Diffusion. In some cases there is only a little amount of data requested from the nodes, and then flooding (as used in DD), is not the best technique to use. An alternative strategy is to flood the network if the number of events is small and the number of queries is large.

In order to flood events through the network the Rumor Routing protocol uses an agent packet. When a node detects an event it adds this event to its event table and generates an agent. Agents travel the network in order to propagate information about local events to distant nodes. When a node generates a query for an event the nodes that know the route may respond to the query by inspecting its event table. Hence there is no need to flood the entire network, which in turn reduces the communication cost. Besides this, the RR protocol only has one path between source and destination, which differs from the directed diffusion protocol. Using this approach, the Rumor Routing protocol can achieve significant energy savings when compared to event flooding and can also handle nodes’ failures.

The disadvantage of the Rumor Routing protocol is that it performs well only when the number of events is small. For a large number of events it is not possible to maintain these agents and event tables.

3.1.5. Maximizing Energy Utilization Routing Protocol (MEURP)

In the literature [92], the researchers propose a flat routing protocol named MEURP. Compared to the previous flat routing protocols, the MEURP provides a waiting approach to relieve the flooding overhead. MEURP also makes use of a multiple routing selection scheme to transmit data. From simulation results, MEURP can improve network lifetime and data packet throughput.
3.2. Hierarchical Routing Protocols

In a hierarchical network the low energy level sensors can be used to perform the sensing task, while the high energy level sensors can be used to process and send data to the base station. Due to this mechanism, energy consumption for the whole network may be decreased. Six hierarchical routing protocols will be introduced as follows:

3.2.1. Threshold-sensitive Energy Efficient Protocol (TEEN)

In TEEN [93] sensor nodes sense the medium continuously, but the data transmission is done less frequently. In this protocol the user can control the trade-off between energy efficiency and data accuracy.

The main feature of this protocol is that if the data that meets the thresholds is not received by the sensors, the sensors will never communicate with each other (conserving energy). In this case, the user cannot receive the data from the network any more. Here the thresholds can be divided into hard threshold and soft threshold. Hard threshold is an absolute value for the sensed attribute. If the node senses this value, it turns on its transmitter and reports the data to the CH. Soft threshold is a small variation in the value of the sensed attribute which causes the node to turn on its transmitter [93].

3.2.2. Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH [94–98] is a TDMA-based MAC protocol (TDMA stands for Time Division Multiple Access. TDMA is a channel access method. This access method can divide the signal into different time slots, thus the users can share the same frequency channel), which is integrated with clustering and a simple routing protocol in wireless sensor networks. LEACH is a hierarchical protocol in which most nodes transmit to
cluster heads and the cluster heads aggregate and compress the data and forward it to the base station.

Each node uses a stochastic algorithm at each round to determine whether it will become a cluster head in a round. LEACH assumes that each node has a radio powerful enough to directly reach the base station or the nearest cluster head, but that using this radio at full power all the time would waste energy.

This thesis will need to complete power control over the radio components on the simulation platform, which in turn makes simulation of power consumption possible. In addition to this LEACH provides sensor networks with many good features, such as clustering architecture and localized coordination.

3.2.3. **Cluster Based Routing Protocol (CBR)**

In the literature [99], the researchers proposed a hierarchical routing protocol named Cluster Based Routing protocol in WSNs. In CBR protocol, the cluster head can receive data from its member during the TDMA time slot and other sensors which just enter the cluster. From the MATLAB simulation results, the authors concluded that CBR protocol reduced the data loss by 25% compared to LEACH protocol.

3.2.4. **Quadrature-LEACH (Q-LEACH)**

In the literature [100], the authors proposed Quadrature-LEACH for homogenous network. In Q-LEACH, network is partitioned into sub-sectors. The clusters formed in these sub-sectors are more deterministic. So, sensors can be well assigned to a specific cluster. From the MATLAB simulation results, this work concluded that Q-LEACH improved stability period, throughput and network lifetime quite significantly.

3.2.5. **Fixed-environment Location-based Clustering Routing Protocol (FLCRP)**

FLCRP proposed in [101] is operated in an environment of fixed sensors only. This
protocol depends on sensor energy and sensor location. FLCRP has setup-phase and steady state-phase. From the simulations, this paper concluded that FLCRP performed better results (packet delivery ratio; average end to end delay) than LEACH.

3.2.6. Mixed-environment Location-based Clustering Routing Protocol

Mixed-environment Location-based Clustering Routing Protocol (MLCRP) proposed in [101] is a modification of FLCRP as it can be used either in fixed or mobile environment. The main difference between MLCRP and FLCRP is the former one does not send any join request packet after receiving the advertisement packet from the cluster heads at the setup phase. From the simulation experiments, this paper concluded that MLCRP performed better results (packet delivery ratio; average end to end delay) than LEACH.

3.3. Location-based Routing Protocols

In location-based routing protocols sensors are addressed by means of their locations. There are two approaches to obtain the location information in this kind of routing protocol. First, the sensor nodes could obtain their location information if each sensor is assigned a GPS receiver. Second, the location information could be obtained by exchanging such information between neighbours. Four examples of location-based protocols are given below.

3.3.1. Greedy Perimeter Stateless Routing (GPSR)

Geographic routing is a routing principle that relies on geographic position information. It requires that each sensor can determine its own location and also the location of other sensors and the sink node. Using this information a packet can be routed to the sink node without knowledge of the network topology. GPSR [52] is such a typical location-based routing protocol, it is a multi-sink protocol for wireless
datagram networks that uses the geographic locations of source nodes, interactive nodes and destination nodes to forward the packet. The GPSR protocol uses the information about a router’s immediate neighbours to forward a packet. If the region to which the GPSR forwards is not accessible this protocol would find another point around the perimeter of the region to forward the packet. As this protocol is an integration of the Greedy and Perimeter protocols it becomes a more accurate and stable routing procedure.

3.3.2. Nearest with Forwarding Protocol (NFP)

The NFP protocol [102] relies on the strategy of greedy forwarding, which tries to bring a transmitted packet closer to the sink node in each step or hop using only local information. Thus each sensor forwards the message to its neighbour that is most suitable from a local point of view. The transmission range or radius of a sensor is the maximum distance a sensor can transmit its data. Progress is defined as the distance between the transmitting sensor and receiving sensor projected onto a line drawn from the transmitter towards the sink node. A sensor is said to be in a transmitter’s forward direction if non-negative progress is produced when the sensor is chosen as the transmitter’s receiver. Using this notion of forwarding for the NFP protocol the most suitable neighbour is then defined to be the nearest neighbour to the transmitter within its transmission range which will result in forward progress. The transmission radius of this sensor is then restricted to the distance to its most suitable neighbour, thus conserving sensor energy and avoiding a data collision. The neighbour can be the sink node itself; if two or more suitable neighbours exist then the packet will be sent to just one using a random choice. Once a path to the sink node from a sensor is established it will remain the path until one of the sensors on the path dies and forces a new path to be found. It is possible that two (or more) sensors could be in a loop, sending packets back and forth between them. Clearly any packets transmitted in a loop will never reach the sink node unless the loop breaks. If a sensor (not in a loop) has no closest neighbours that make forward progress and is unable to broadcast directly to
the sink node, then the packets from that sensor will also never reach the sink node. However, no new sensors can be added into the network. The reason for this can be explained as follows: In the experiments of this thesis, sensors are randomly placed in the simulation area. When the deployment is done, the sensors are static. The protocols operate on static network only. Adding or removing sensors will change the routing strategies and this research has not been taken into account in this thesis. As with any greedy strategy, the NFP protocol does not in general produce an optimal network routing, but instead one that may approximate such a routing in a reasonable time.

3.3.3. **Nearest Closer Protocol (NC)**

The NC protocol relies on the strategy of greedy forwarding as well, which tries to bring a transmitted packet closer to the sink node in each step or hop using only local information. The difference between NC and NFP is that in NC the distance between the receiver sensor node and the sink node is shorter than the distance between the transmitter sensor and the sink node, instead of forward progress as in NFP. Thus put simply in NC the transmitter sensor will transmit to its nearest neighbour that is closer to the sink node. The difference between NFP and NC is illustrated in Figure 3.1.

![Figure 3.1: Greedy Routing Strategies](103).
In Figure 3.1 the neighbours A, B, C, E, and F are in the forward direction from S. (This thesis will not refer to A, B and F any further.) The remaining node G is in the backward direction from S. Here S stands for sender and D for destination. Hou and Li [102] proposed Nearest with Forward Progress, where each node sends the packet to the nearest neighbour with forward progress (e.g. node E in Figure 3.1). Stojmenovic and Lin [104] defined Nearest Closer, which is a modification of NFP considering distance instead of progress, i.e. packets are forwarded to the nearest neighbour among all neighbours closer to the destination (e.g. node C in Figure 3.1).

3.3.4. Dijkstra-based Localized Energy-efficient Multicast Algorithm (DLEMA)

DLEMA is proposed in the literature [105]. DLEMA focuses on discovering energy shortest paths leading through sensors to the destinations. Simulation experiments demonstrate that DLEMA provides low delays and high success rate. Thus, the researchers conclude that DLEMA can be considered as a solution for geographic dissemination of multimedia streams in complex sensor networks.

For location-based protocols, many researchers select latency as the key parameter to evaluate the protocol. The term latency refers to the time taken by a packet to be transmitted across a network from source node to destination node that includes all possible delays caused during route discovery, retransmission delays at the MAC layer, propagation. A protocol with high latency means the performance of the protocol is bad due to network congestion and *vice versa*.

Chapters 2 and 3 detailed several routing protocols based on different overall strategies. Table 3.1 summarizes recent research results on data routing in WSNs using the afore-mentioned routing protocols.
Table 3.1: Classification and Comparisons of Routing Protocols in WSNs.

<table>
<thead>
<tr>
<th></th>
<th>Classification</th>
<th>Scalability</th>
<th>Network lifetime</th>
<th>Mobility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-hop</td>
<td>Flat</td>
<td>Limited</td>
<td>Very Good</td>
<td>Limited</td>
</tr>
<tr>
<td>SPIN</td>
<td>Flat</td>
<td>Limited</td>
<td>Good</td>
<td>Possible</td>
</tr>
<tr>
<td>DD</td>
<td>Flat</td>
<td>Limited</td>
<td>Good</td>
<td>Limited</td>
</tr>
<tr>
<td>RR</td>
<td>Flat</td>
<td>Good</td>
<td>Good</td>
<td>Limited</td>
</tr>
<tr>
<td>MEURP</td>
<td>Flat</td>
<td>Limited</td>
<td>Very Good</td>
<td>Limited</td>
</tr>
<tr>
<td>TEEN</td>
<td>Hierarchical</td>
<td>Good</td>
<td>Good</td>
<td>Fixed sink</td>
</tr>
<tr>
<td>LEACH</td>
<td>Hierarchical</td>
<td>Good</td>
<td>Very Good</td>
<td>Fixed sink</td>
</tr>
<tr>
<td>CBR</td>
<td>Hierarchical</td>
<td>Good</td>
<td>Very Good</td>
<td>Fixed sink</td>
</tr>
<tr>
<td>Q-LEACH</td>
<td>Hierarchical</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Fixed sink</td>
</tr>
<tr>
<td>FLCRP</td>
<td>Hierarchical</td>
<td>Good</td>
<td>Good</td>
<td>Limited</td>
</tr>
<tr>
<td>MLCRP</td>
<td>Hierarchical</td>
<td>Good</td>
<td>Good</td>
<td>Possible</td>
</tr>
<tr>
<td>GPSR</td>
<td>Location</td>
<td>Limited</td>
<td>Low</td>
<td>Limited</td>
</tr>
<tr>
<td>NFP</td>
<td>Location</td>
<td>Limited</td>
<td>Low</td>
<td>Limited</td>
</tr>
<tr>
<td>NC</td>
<td>Location</td>
<td>Limited</td>
<td>Low</td>
<td>Limited</td>
</tr>
<tr>
<td>DLEMA</td>
<td>Location</td>
<td>Limited</td>
<td>Low</td>
<td>Limited</td>
</tr>
</tbody>
</table>

3.4. Reasons for the selection of routing protocols

There are a number of reasons for choosing the three routing protocols considered in detail in this thesis. The choice of the Single-hop routing protocol from the flat category will provide a baseline [106] to which other routing protocols can be compared. The Single-hop routing protocol is the most simple of all routing protocols and therefore it should be comparatively easy to build a mathematical model of the results of the simulations. The Single-hop routing protocol is also used in wireless sensor networks where communication is just between the sensor node and the sink.
node (no intermediate sensors) and it can be implemented with inexpensive sensors. The most well-known and widely-used hierarchical routing protocol is LEACH. The results from all hierarchical protocols are more difficult to model mathematically, because sensors perform different roles at different times. In LEACH these roles are limited to two: acting as a normal sensor or acting as a cluster head, which makes the modeling task feasible. Thus the choice of LEACH, is mainly based on the fact that it is widely-used. Finally, the location-based routing protocols, of which there are a large number, but difficult to be implemented in J-Sim Greedy Perimeter Stateless Routing (GPSR) essentially allows backtracking if a dead end is reached, so within any implementation program for GPSR and other similar routing protocols there must be a large number of nested conditional statements of the ‘IF’ and ‘THEN’ form. This means that the behaviour of the protocol can fundamentally change during a simulation depending on the geographical and/or power status of the network, so that trends in results obtained from simulations using such protocols will be difficult (if not impossible) to analyze. The choice in this thesis, to avoid such conditionality and the apparent complexity of any resultant mathematical model, came down to Nearest with Forward Progress (NFP) or Nearest Closer (NC) for the work. The routes in NFP along which sensors transmit data to the sink node will be very jagged (with lots of ‘ups’ and ‘downs’) compared to those in NC by definition. Thus NC presents, at face value at least, a more sensible approach to transmitting data from a sensor to the sink node and hence our choice of it as a representative for location-based protocols.

The reason why routing protocols are being divided into the three categories can be found in the literature [107]. It may be the case that some hybrid routing protocols cross the boundaries between the three main types of routing protocols. Once again the behaviour of such a hybrid will be conditional upon the state of the wireless sensor network, so that during a simulation it could act repeatedly as a flat, hierarchical-based or location-based routing protocol depending on the circumstances.
The simulation results from a routing protocol that allows multiple and fundamental changes of behaviour, will be virtually impossible to model mathematically without effectively reverse engineering the protocol implementation program.

3.5. Summary

This chapter has discussed three different categories of routing protocols, for each category a number of examples have been considered and described. In future chapters this thesis has selected a subset of the routing protocols listed in this chapter to analyze as three representative routing protocols: Single-hop, LEACH and Nearest Closer are taken as being a representative of flat, hierarchical and location-based routing protocols respectively.

Prior to commencing simulations the next chapter will define the significant evaluation parameters to be considered and utilized within such simulations.
Chapter 4: Evaluation Parameters

This chapter will define the evaluation parameters previously mentioned in Section 2.4.

Firstly, the basic evaluation parameters of Reliability (R) and Lifetime (L) will be defined for use in the next three chapters. In particular, the Lifetime will be used as a measure of energy consumption. This thesis will also focus on varying the parameters of Radius (Ra) and Density (D), which are also defined below. The other parameter that is defined here is that of Latency (L), although this parameter is only considered in passing in this work. Based on the results of experiments (in Chapter 5, 6 and 7) of varying these last three parameters and measuring the effects on the Lifetime and Reliability, evaluation models will eventually be constructed.

This thesis has adopted power control over the radio components on the J-Sim, which makes simulation of power consumption possible.

In this thesis it is assumed that all sensors can transmit data about the phenomena they detect and can receive and retransmit data from other sensors about any of the phenomena detected in the system.

4.1. Evaluation parameters

4.1.1. Reliability

In this thesis, experiments are conducted using the concept of Reliability, defined by:
\[
\text{Reliability} = \frac{\text{the number of packets received by the sink node}}{\text{the number of packets sent directly to the sink node}}
\]  

The advantages of using Reliability \([108]\) as defined are two-fold. Firstly, this definition can be used in the field and laboratory. In the field, provided the sink node (end-user) knows how many cluster heads or sensors there are and the average number of packets of sensed data per sensor per time unit, then the Reliability can be estimated at any time during the lifetime of the network using the number of time units elapsed. Any difference between the Reliability measured in the field and laboratory could be used to detect how many packets of data are lost between sensors and cluster heads or the sink node (This would include counting sensors that have failed or have otherwise been lost to the system). Secondly, this definition makes it easy to analyze communication among the sensor nodes; in particular it makes the estimation of data collision in the wireless sensor network possible.

4.1.2. Lifetime

Network lifetime has become the key characteristic for evaluating sensor networks in an application-specific way. Network lifetime \([109]\) is the time span from the deployment of the sensors to the instant when the network is considered non-functional. The time for non-functional could be the instant when the first sensor dies, a loss of coverage occurs, or a percentage of sensors die. The definition for network Lifetime which has been taken in the experiments in this thesis is the time when the last packet is received by the sink node. The reason for selecting this definition of Lifetime is because a sensor network doesn’t make sense if the sink node cannot receive any packet. (In the experiments in the Single-hop and LEACH protocols, the last sensor may maintain the transmission rate of packets until it cannot transmit data to the sink node. In this case, the simulation should wait until the sink node cannot receive any information. However, in the experiments in the Nearest Closer Protocol, some sensors may be alive after the sink node stops receiving data. In
this case, if the data transmission process is blocked, the sensors will continue to work for a long time until they run out. So to choose the time when the last sensor is dead as the definition for network *Lifetime* is not a good choice.)

4.1.3. **Radius**

In a wireless sensor network, all the nodes are deployed with a predefined communication radius [110]. In each experiment of this thesis, all the common sensors will have the same communication radius. This limits the area that the sensors can detect phenomena in, so that this thesis will also evaluate the effect that this limitation on communication radius has on the *Lifetime* and *Reliability* by varying the *Radius*.

4.1.4. **Density**

Density is a parameter for researchers to evaluate routing protocols. As the density of sensor nodes increases, the efficiency of the sensor network may improve or deteriorate. So in some of the simulations researchers will be concerned about the number of sensors they are using in the network. It is obvious that some of the nodes will be out of power in a real wireless sensor network. Thus attention will naturally turn to the number of sensors that are still alive. The number of sensors deployed in a fixed area will be taken as the *Density* parameter [111] in this thesis. Obviously as the number of sensors goes up, so does the average number of sensors per square metre i.e. the density of the sensors.

4.1.5. **Latency**

*Latency* in this thesis indicates the (average) number of hops from a sensor to the sink node or cluster head. In general latency refers to the time taken for data to be transmitted to the sink node, each hop that is needed involves a packet of data to be received, processed and forwarded along its transmission path.
In the rest of this thesis the terms Reliability, Lifetime, Radius, Density and Latency written with a capital initial letter and in italics will refer specifically to the definitions made in this chapter.

4.2. Experimental Set-up

In Figure 4.1, the simulated area for the following experiments is defined as a 10 metre × 10 metre square with randomly deployed sensors. The sink node for this application is located in the middle of this area. One of the primary reasons for selecting this set-up is to allow the results to be generalized to large areas by concatenation of networks similar to this. For example, a 50 metre × 50 metre region could be configured using 25 instances of the set-up used here in a 10 ×10 grid formation. (Expansion to 50 metre×50 metre would require 25 sink nodes. Each sink node is located in the centre of a separate 10m x 10m square. In this case, there are 25 small squares. The sensors nodes are still randomly placed. The transmit radius is the same as before. This theory was introduced in the literature [84], but the theory ignored the effect of data collisions among small squares. This is a potential weakness of this approach.) There exists a target node in each simulation experiment, and this target node will generate a stimulus every one second. All the sensors are active at the beginning of the simulation, and the simulation time is enough for obtaining data in each separate experiment. In the experiments of Chapter 5, 6 and 7 (Section 5.3, Section 6.4, Section 7.2), this thesis will vary the number of sensors and radius to see effect on Reliability and Lifetime. In Chapter 6 (Section 6.4), the experiments will vary the number of clusters to see effect on Reliability and Lifetime. All the points (in the figures) in this thesis are the average value from 5 separate experiments. In each experiment the sensors are static once placed, and the sensor nodes are re-positioned randomly for the 5 experiments. The Network Lifetime will be measured in seconds. The maximum simulation time for all experiments in this thesis is 100,000 seconds.
Details of the parameters will be described in Chapter 5, 6 and 7.

This model is applicable in many instances, including thermal radiation, light, sound, magnetic and gravitation fields. Under this set-up, all of the nodes remain active and no hibernation of any nodes takes place.

In Figure 4.1, the circles stand for cluster heads (for LEACH), the diamonds for normal sensors, and the triangle for the sink node.

This thesis will choose a suitable MAC protocol for each routing protocol. For the Single-hop protocol, Carrier Sense Multiple Access (CSMA) has been chosen as the MAC layer protocol as it can provide a mechanism to manage communication between sensors, which will avoid data collisions. For the LEACH protocol, Time-Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) have been selected as the MAC protocols as TDMA has a natural advantage of collision-free medium access and CDMA can allow sensors from different clusters to
transmit their data at the same time without a data collision occurring. For the Nearest Closer protocol, Slotted ALOHA MAC protocol has been selected as it can make a single active sensor nearly continuously transmit at full channel rate, thus better results can be obtained for the NC.

4.3. Summary

This chapter has defined the evaluation parameters of Lifetime and Reliability (L and R) and the variable parameters of Radius (Ra), Density (D) and Latency (L). In the following chapters, the Reliability and Lifetime will be evaluated using Density and Radius as variable parameters. As the number of sensors increases in a fixed space the Reliability and Lifetime of the sensor network may improve or disimprove, thus in some simulations necessarily the density of the sensors in the network will have to be considered. Of course, in a real wireless sensor network, some of the nodes will have run out of power, so in this situation it will be necessary to focus on the number of live sensors.

After the analysis of evaluation parameters in this chapter, the subsequent chapter will focus on the simplest flat routing protocol, namely that of the Single-hop protocol.

A large number of sensor nodes will consume a lot of energy, consequently the sensor network will require network routing protocols which provide network control and management functions. The simplest of the so-called flat protocols, namely the Single-hop protocol will be analyzed in this chapter. The reason for studying this well-known protocol is to provide a base-line, so that comparisons can then be made between this protocol and the more complex ones considered in the following two chapters.

5.1. The Single-hop Protocol

In this chapter the relationships among Radius, Density, Lifetime and Reliability will be investigated for the Single-hop protocol by simulation. The Single-hop routing protocol mentioned in Section 3.1.1, also known as the direct method or the one-hop method, has all sensor nodes communicating directly back to the base station. This model is normally not feasible for most applications, because of severe restrictions and inefficiencies. Its main fault is that sensors are traditionally equipped with relatively low power radios limiting the distance to which they can communicate. Therefore, any sensors that are located at a greater distance than the communication radius or that have more obstacles between them and the sink node (i.e. no line of sight) will have higher SNR or possibly not be able to reach the base station.

The Single-hop protocol also incorporates a MAC layer. The MAC layer is based on MAC-CSMA, a custom MAC protocol based strictly on Carrier Sense Multiple Access (CSMA).
In this non-persistent CSMA protocol, before sending the data, the station senses the channel and if the channel is idle it starts transmitting the data. On the other hand, if the channel is busy, the station does not continuously sense it, but instead waits for a random amount of time and repeats the algorithm.

5.2. System Architecture and Experimental Set-up

In order to deliver the application this thesis adopted a standard protocol stack. When different sensors want to communicate, they cannot do so at the same time due to interference on the channel, so a MAC layer is required in order to mediate the use of the channel and to retransmit failed packets.

J-Sim is a Java-based simulation tool and it also provides the 802.11 to all users. The MAC layer will typically use an RTS/CTS [112] (Request To Send/ Clear To Send is an additional method to implement virtual carrier sensing in Carrier Sense Multiple Access with collision avoidance) mechanism to manage communication between sensors, but these control messages may be lost due to data collision.

The experimental set-up in J-sim has been mentioned in Section 4.2. The routing protocol is Single-hop. As the sensors are randomly placed in the simulation area, the position of the nodes will change in different simulation, which will have an effect on the results if the number of sensors is limited.

5.3. Results

5.3.1. Density—Lifetime—Reliability
A series of experiments was carried out with the number of sensors starting at 10, and increasing in increments of 10 to 60. The transmission radius for each sensor and the
target node was fixed at 15 metres.

![Graph showing density and reliability relationship](image)

Figure 5.1: Density, Reliability Relationship for Single-hop.

In Figure 5.1, the relationship between the number of sensors and the Reliability is very clear. The Reliability for this application decreased as the number of sensors increased. Although the curve is essentially a straight line, the Reliability is above 99% in all cases considered. Using the least-squares approximation from statistics, the red line was fitted through the six data points in Figure 5.1. The equation of this line with \( n \) representing the number of sensors is

\[
100.014 - 0.0126857142857145n \tag{1}
\]

This decreasing graph can be explained by observing that as the number of sensors goes up in the fixed space more sensors will join the data transmission process, and
consequently the communication at the sink node will become more and more complex. Since a large radius is being used all sensors can communicate with the sink node and therefore the only packet losses that can occur are through data collision. The CSMA protocol is designed to minimize such collisions, but at the expense of increased latency with increased number of sensors. The results show that the protocol has a small error percentage associated with it (equal to the coefficient of \( n \) above), allowing data collision to occur and this explains why the Reliability is inversely proportional to the Density.

![Figure 5.2: Density, Lifetime Relationship for Single-hop.](image)

In Figure 5.2, the Lifetime reached its lowest value when the number of sensors equalled 40, whereas the highest value of the Lifetime occurred when there were 20 sensors. All in all, the Lifetime is between 14900 and 16100 seconds. The reason why the Lifetime is around 15600 is that any sensor in this network could transmit data to
the sink node directly; that is to say almost all the sensors do not have enough energy to transmit packets to the sink node when the first sensor fails to transmit packets to the sink node. Thus, the change for *Lifetime* is less than 10% as the number of sensor nodes increases.

The actual value obtained for the constant *Lifetime* will of course vary (better batteries may increase *Lifetime*) depending on the battery size and power available to all sensor nodes, which is assumed to be the same for each sensor in our experiments. If the battery size and power was a variable per sensor then the conclusions of this experiment, of a constant *Lifetime*, could no longer be drawn.

### 5.3.2. Radius—*Lifetime*—*Reliability*

A series of experiments was carried out with the transmission radius starting at 1 metre and increasing by one-metre increments to 10 metres. The number of sensors was fixed at 20 (The effect of data collision at the sink node is weak. Clearly as the number of sensors increase in the square, the average distance between sensor decreases (these average distances are given in Table 7.5), as does the average distance(s) of the nearest sensor(s) to the sink node. A small number of sensors was chosen here to guarantee that the distance of the nearest sensor to the sink node isn't too small. This means that increasing the transmission radius will have some effect on *Reliability* and *Lifetime* as demonstrated. Using, say 100, sensors will undoubtedly just result in no variation in *Reliability* and *Lifetime* by increasing the *Radius* in one metre increments, which would negate the point of the experiment. Indeed even in the 10 experiments *Lifetime* essentially becomes constant for a transmission radius of two metres or above.).
Figure 5.3: Radius, Reliability Relationship for Single-hop.

Figure 5.3 shows the relationship between the Reliability and the transmission radius. It is necessary that all the sensor nodes adopt the single-hop transmission technique to deliver messages. The Reliability for this application increased as the radius increased from 1 to 7 metres. The Reliability reached its highest value when the radius equalled 7 metres. As the radius increased from 7 to 10 metres, the Reliability value can no longer increase, as the radius is sufficiently large to transmit the data from anywhere in the square to the sink node in the centre.
Figure 5.4: **Radius, Lifetime Relationship for Single-hop.**

Figure 5.4 shows the relationship between the *Lifetime* and the transmission radius. The overall trend for this curve was that it was a horizontal line provided the radius was greater than two metres. The reason for this curve could be explained as follows: when the transmission radius is equal to one metre, no sensor could communicate to the sink node due to the low radius in some of the experiments; however, there exist one or more sensors that could transmit packets to the sink node in the other experiments. Consequently the average *Lifetime* value for a one-metre transmission radius is 5718 seconds. However, when the transmission radius is equal to two metres or more, at least one sensor could transmit packets to the sink node directly in all experiments, so the trend of the *Lifetime* is essentially a horizontal line for these larger radii.
5.4.  **Reliability and Lifetime Models**

5.4.1.  **Fixed Radius Model**

The relationship between the number of sensors and the Reliability for a fixed 15-metre transmission radius was found in Section 5.2.1 (formula 1) to be

\[ 100.014 - 0.0126857142857145n \]

where \( n \) represents the number of sensors. The value of 100 should be obtained when \( n = 1 \), whereas the equation yields 100.001343 indicating that this linear model is a good fit to reality. On the other hand, the Reliability is 0 under this model when \( n = 7,884 \), although it is perhaps dangerous to extrapolate so far on the limited experimental data obtained.

A very simple Lifetime model can be built by assuming that a sensor starts with a total battery lifetime of \( K \) and the sensor consumes \( e \) on average per second (transmitting with a fixed radius). Then the network will have a fixed Lifetime of \( K/e \) independent of the number of sensors. For the experiments conducted with a radius of 15 metres the average value of \( K/e \) was 15600.

To justify this model in J-Sim the energy consumed by a sensor is proportional to the square of its distance \( d \) to the centre (detailed in Section 2.3.2). It is a fact that randomly distributing \( n \) sensors in the square and averaging the distance (or its square) to its centre is the same as randomly distributing just one sensor and averaging its distance (or its square) over \( n \) repetitions of the experiment. It follows that \( E(d^2) \) averaged over all sensors in the network measured over a large number of experiments is independent of the number of sensors and is approximately equal to 16.6 metres. The Lifetime of the network should thus be modelled by
where $\alpha$ is a constant. So $e = 16.6\alpha$ or equivalently $K = 258960\alpha$.

5.4.2. Varying Radius Model

As the transmission radius $r$ varies, the expected number of sensors within range will be given by

$$20\pi r^2/100$$

for $0 \leq r \leq 5$.

This yields the answers $0.62831853$, $2.513274123$, $5.654866776$, $10.05309649$, and $15.70796327$ for $r = 1, 2, 3, 4$ and 5 metres respectively, and obviously the answer is 20 for $r \geq 5\sqrt{2}$ metres. For $5 < r < 5\sqrt{2}$, some elementary geometry yields that the area within the square but outside the circle centred at the sink node is given by

$$A = 2\left(5 - \sqrt{r^2 - 25}\right)^2 + 2(50 - r^2) - 4\left(\frac{\pi}{4} - \theta\right)r^2$$

where $\theta = \tan^{-1}(\sqrt{r^2 - 25}/5)$. Thus the expected number of sensors in this case is $20 - A/5$, which gives $19.016$ for $r = 6$ metres.

The sensors in the Single-hop network considered remained active all the time. A sensor within range of the sink node can adjust its transmission range to the distance to the centre and thus conserve power. On the other hand each sensor out of range transmits to the full transmission radius. Out-of-range nodes still transmit is to be consistent with the later experiments, so that the only behavioral change from
experiment to experiment is the routing protocol. The transmissions of sensors out of range are not controlled by the CSMA protocol and thus in particular can cause a data collision with data transmitted by sensors within range, thus adversely affecting the Reliability. A study of the raw data from the experiments reveals that there is, as one might expect, a linear relationship between the number of sensors within the transmission range, $s$, (for $r \leq 5$) and the number of packets received by the sink node. More surprisingly there is also a linear relationship between $s$ (for $r \leq 5$) and the number of packets sent to the sink node.

The data obtained was averaged out for constant values of $s$ for different values of $r$, and then using the least squares method the total number of packets transmitted is approximately

$$206230.002000859 - 9193.82884070416s$$ (5)

for $0 \leq r \leq 5$. Also using the same method the number of packets received by the sink node is approximately

$$436.736796908544 + 1280.42582653499s$$ (6)

for $0 \leq r \leq 5$. Thus on average the Reliability for 20 sensors and $0 \leq r \leq 5$, may be approximated by the formula 7.

$$\frac{436.736796908544 + 1280.42582653499r^2}{206230.002000859 - 9193.82884070416r^2}$$ (7)

This formula 7 generally over-estimates the Reliability figures obtained in the experiments conducted, but has the correct general shape. The graph of the function
represented by the formula above (the red curve in Figure 5.5) has been overlaid with that of the Reliability in Figure 5.3 (in blue in Figure 5.5).

Figure 5.5: A Model of the Reliability.

Now if the number of sensors is fixed at 20 and $r \geq 5\sqrt{2}$ metres, then the basic Reliability figure from the linear equation stated at the start of Section 5.3.1 is 99.76%.

The best formula that can be proposed for $5 < r < 5\sqrt{2}$, is that the Reliability is given by

$$\left(1 - \frac{A}{100}\right)(0.9976) \quad (8)$$

where $A$ is the area found above.
Finally we consider the *Lifetime* of the network as the radius varies. The experiments show that the average *Lifetime* is constant at about 15600 (it is actually 15587) independent of the radius, provided there is at least one sensor that can successfully transmit to the sink node. Now the expected square of the average distance of sensors within the transmission range will increase with the radius up to $5\sqrt{2}$ metres, but as calculated above, the expected number of sensors within range increases in proportion to the square of the radius for $0 \leq r \leq 5$. Thus for a smaller radius the fewer sensors within range will transmit more than in the case of a larger radius. The experiments conducted suggest that the transmission rate is proportional to the square of the distance of the sensor from the sink node, so that the energy consumption and the transmission rate effectively negate each other leading to a constant *Lifetime* (The transmission rate is the number of packets sent per second, the energy consumed is the energy needed to send one packet to the sink node. So the transmission rate times energy consumed per packet gives the energy consumed per second (this is the $e$ of Section 5.4); dividing $K$ (the total battery lifetime) by $e$ gives the *Lifetime*).

The probability, $p$, that there are no sensors within the transmission range is given by

$$\left(1 - \frac{E(s)}{20}\right)^{20}$$

(9)

where $E(s)$ is the expected number of sensors within the transmission range calculated above. So the expected *Lifetime* is given by $15600(1 - p)$. This formula yields values of (almost exactly) 15600 for $r \geq 3$ and is given by

$$15600\left(1 - \left(1 - \frac{\pi r^2}{100}\right)^{20}\right)$$

(10)
for $0 \leq r < 3$. In particular this formula yields 7361 and 14537 for $r = 1$ and 2 metres respectively. The graph of the function represented by the formula above (the red curve in Figure 5.6) has been overlaid with that of the *Lifetime* in Figure 5.4 (in blue in Figure 5.6).

![Figure 5.6: A Model of the Lifetime.](image)

In the experiments of this chapter, the sensors out-of-range still need to transmit the packets to the sink node (This process will consume a lot of energy). These sensors can be regarded as redundant sensor nodes. So there is no need to save energy for these redundant nodes. The data collision may occur, but the effect for the *Reliability* is weak. If the redundant sensors cease transmission, the whole simulation time will be increased a lot. In future, if a sensor knows its distance from the sink node, and this distance is beyond its range, then the sensor will not need to transmit the data. In this case, the *Reliability* can be increased. This applies to the LEACH protocol (in Chapter
6) and the Nearest Closer protocol (in Chapter 7).

5.5. Summary

In this chapter some useful results among the Density, Radius, Reliability and Lifetime parameters have been obtained.

As the number of sensors increased from 10 to 60 the Reliability decreased. The Reliability is above 99% in all cases considered. The Lifetime is between 14900 and 16100 seconds as the number of sensors increases from 10 to 60.

The Reliability increased as the radius increased from 1 to 7 metres. As the radius increases from 7 to 10 metres, the Reliability can no longer increase. The Lifetime is essentially a horizontal line when the radius is equal to 2 or greater than 2 metres.

The results thus demonstrate (at least in the experimental set-up) that the Lifetime is not affected by the number of sensors (assuming this number is not very small or large). It may be the case that the Lifetime is being predominantly measured by the Lifetime of the nearest sensor to the sink node (which should have the least energy expenditure). Increasing the density of sensors will on average decrease the distance of the nearest sensor to the sink node but this decrease will be very marginal in the range of 10 to 60 sensors and so will only have a very minimal effect on Lifetime.

Intuitively one would expect that as the number of sensors increases that the number of data collisions will increase and hence that Reliability will fall, as is the case. Given that this is a very simple protocol the relationship between the number of sensors and Reliability is also direct in that it is linear. Perhaps more surprising is that the Reliability is so high in all the cases considered. It is the case in any network protocol
that a sensor won’t transmit if it knows that another signal is present on the network
(this relies on feedback from the sink node) and it appears to be the case that this very
simple protocol implementation works extremely efficiently in our experiments.

The \textit{Lifetime} appears to be measured in Single-hop predominantly on the sensor
nearest to the sink node. This distance is just slightly larger than 1.5985 for 20 sensors
from Table 7.5 and remarks in Chapter 7. This graph shows that for a radius of 2
metres or more the nearest sensor is essentially always closer than the radius to the
sink node, whereas for a radius of 1 metre it is averaging out the \textit{Lifetime} when the
nearest sensor is within one metre (which should be quite high) with the cases when it
isn’t (0 \textit{Lifetime}). So in a sense the figure on Figure 5.6 for one metre is almost a
(unquantified) frequency measure of how often the nearest sensor is within one metre
of the sink node.

\textit{Reliability} is measured over all sensors transmitting to the sink node, and it is clear
that with a small radius that a number of sensors won’t be able to successfully
transmit to the sink node thus adversely affecting \textit{Reliability}. Once 7 metres is
reached the transmission radius is no longer relevant (as all sensors can transmit to the
sink node) and the only drain on \textit{Reliability} will be data collisions, which will be on
average constant for a fixed number (20 here) of sensors.

Furthermore two valuation models amongst the parameters of \textit{Lifetime}, \textit{Reliability} and
\textit{Density} have been proposed. Based on these intelligent evaluation models, wireless
sensor network users can predict the \textit{Lifetime} and \textit{Reliability} directly. This means that
sensor nodes can be deployed in such a network without further simulations.

After the analysis of the simplest flat routing protocol, Single-hop, the subsequent
chapter will focus on one of the typical hierarchical-based routing protocols, namely
LEACH. For this protocol some J-Sim simulation results and evaluation models will also be proposed.
Chapter 6: Evaluation Models for the LEACH Protocol

Based on the definitions provided in Chapter 4, some *Lifetime, Density, Radius* and *Reliability* analysis has been undertaken for the LEACH protocol. Evaluation models among the parameters of *Lifetime, Density, Radius* and *Reliability* are constructed. These evaluation models mean that wireless sensor network users can predict the *Lifetime* and *Reliability* based on LEACH directly. Thus, simulation tools will not be necessary for the deployment of sensors resulting in a saving of both time and money.

6.1. LEACH Protocol

LEACH (mentioned in 3.2.2) is a TDMA-based MAC protocol, which is integrated with clustering and a simple routing protocol in wireless sensor networks.

The LEACH protocol differentiates itself from other cluster-based routing protocols in that:

(a) It uses randomized rotation of the cluster heads;
(b) It reduces the amount of data that needs to be transmitted to the sink node;

The use of clusters decreases the number of intermediate nodes. Furthermore, using rotating cluster heads and adaptive clusters, the energy requirements of the system are in general distributed among all the sensors.

LEACH is one of the hierarchical protocols in which most sensor nodes transmit
sensed data to cluster heads; the cluster heads aggregate and compress the data and in turn forward the data to the sink node. Each node uses a stochastic algorithm at each round to determine whether it will become a cluster head in that round. This algorithm can be explained as follows:

The decision of node $n$ to become a cluster head or not is made by choosing a random number between 0 and 1. If the number is less than a threshold $T(n)$, the node becomes a cluster head for the current round. In fact $T(n)$ is set equal to $P/ (1 − P * (r mod (1/P)))$, if node $n$ belongs to the set of sensor nodes that have not been cluster heads in the last $1/P$ rounds and otherwise it is set equal to 0. Here $P$ (with $1/P$ an integer) is the desired percentage of cluster heads and $r$ stands for the current round, see [94] for more detail.

Consequently, if the remaining energy for each node can be measured, it will make a significant contribution to this research area.

LEACH is one of the representative hierarchical protocols, and LEACH provides sensor networks with many good features, such as clustering architecture and localized coordination. In addition, LEACH is the simplest hierarchical protocol to be implemented in J-Sim, which will cut down on simulation time. Thus this thesis analyzes some critical parameters based on the LEACH protocol within J-Sim.

6.2. Loss of Data

There are three primary contributors to the dropping of data packets, these are:

(a) A sensor may receive data from the target, but be unable to transmit it to any cluster head through lack of power or insufficient $Radius$.

(b) Aggregated data sent from cluster heads can be dropped through data
collision with packets from other cluster heads.

(c) Aggregated data sent from cluster heads can be dropped through lack of power or insufficient Radius.

The definition of Reliability excludes data dropped in the first way and so the primary concern in this thesis should be data collision and the lack of power of cluster heads. A data collision is the simultaneous presence of signals from two nodes on the network. A collision can occur when two nodes each think the network is idle and both start transmitting at roughly the same time. When a data collision occurs, all the data from the two (or more) sources are lost.

6.3. System Architecture and Experimental Set-up

Considering a single cluster in isolation, the use of TDMA (detailed in Section 2.1.2) ensures that no data collision occurs within the cluster. The LEACH protocol also employs Code Division Multiple Access CDMA [113–115], this uses spread-spectrum technology and a special coding scheme to allow sensors from different clusters to transmit their data at the same time without a data collision occurring. However, data from different cluster heads can collide en route to or at the sink node leading to a severe loss of data.

All the experiments conducted within this thesis are based on the J-Sim (mentioned in 2.4) simulation tool. The routing protocol is LEACH. The simulated area for the following experiments is defined as a 10 metre by 10 metre square with randomly deployed nodes. The sink node for this application is located in the centre of this area.
6.4. Results

6.4.1. Density—Lifetime—Reliability

In order to analyze the effects of Density, the network Lifetime and Reliability on WSN performance, a series of experiments was undertaken varying the number of sensor nodes starting at 2, increasing to 4, 6, 9, 12, 15, 18, 21, 24, 30, 45 and finally 60. (The number of sensors in the single-hop model were taken to be 10, 20, 30, ..., 60 and it was clear following these experiments that Lifetime was essentially constant and Reliability was linear. For the LEACH case it was necessary to conduct more experiments to demonstrate the curves that arose from the experiments.) The transmission Radius for each sensor remained at 10 metres and the number of clusters for this application was two (Having two cluster heads is the easiest situation to analyze from the viewpoint of data collisions at the sink node meaning that an intuitive insight to what occurs can be applied. Nevertheless varying the number of clusters is considered in Section 6.4.3). The transmission Radius for the target node was also set at 10 metres. The transmission rate of a sensor is fixed in this experiment, which means sensors will transmit at the same rate as the others.

As the number of sensor nodes increases in the fixed space, more sensors can be elected as cluster heads and so the energy for each sensor can be conserved. Thus the Lifetime of the network should increase as the number of sensors goes up. Figure 6.1 confirms as expected that the network Lifetime increases as the Density increases. In other words, the relationship between the Density and the network Lifetime is that of positive correlation.
Figure 6.1: (a) Number of Sensors, *Lifetime* Relationship. (b) Number of Sensors, *Reliability* Relationship.
The Lifetime for this experiment in Figure 6.1 (a) increases from 1541 to 2097, 2503, 2779, 3077, 3307, 3427, 3475, 3601, 3707, 3903 and finally to 4043. Figure 6.1 (b) shows the relationship between the Reliability and the number of sensors. The Reliability for this experiment decreased from 22.86% to 16.35%, 12.44%, 9.31%, 7.46%, 6.45%, 5.13%, 4.67%, 4.38%, 3.23%, 2.40% and finally to 1.89%. Thus the relationship between the number of sensors and the Reliability is one of negative correlation. This may be explained as follows: As the Density increases, the cluster heads will consume much more energy to communicate with the sensor nodes. Then the remaining energy for each cluster head will decrease rapidly with the increasing Density. In addition, transmitting data to the sink node will consume a lot of energy, and then more and more elected cluster heads cannot transmit data to the sink node. Thus it is reasonable to expect the Reliability to decrease with the Density.

When the number of sensors equalled 60, the Reliability was found to be 1.89%. Compared to the previous data (when the number of sensors equals to 10 to 50), the Reliability has reached a very low level, but at this point, the network Lifetime reached the highest value 4043. Thus it is possible for users to choose an optimum Density value for this application depending on the Reliability required.

The Lifetime for LEACH is much lower than that for Single-hop. This is because cluster heads will consume energy in communicating with the sensors in the clusters, aggregating data and in transmitting aggregated data to the sink node. Although in LEACH this role is rotated it results in high energy expenditure. For example if we have two clusters with thirty sensors, then each cluster head has to deal with 15 sensors roughly. Our model shows that each cluster head expends three times energy of one normal sensor per time unit, which means that the network expends more energy per unit time compared to not using cluster heads leading to a reduced Lifetime. It is also the case that Lifetime for Single-hop is being predominantly measured on the Lifetime of the nearest sensor to the sink node, which will very long. The ratio of
Lifetime for Single-hop versus LEACH in these experiments is about 4 for 60 sensors.

The Reliability for LEACH is much lower than that for Single-hop as well. This is because cluster heads will consume a lot of energy to communicate with the sensors in the clusters. The cluster heads may not have enough energy to transmit packets to the sink node, but they will continue to run and drop packets. Also when data collisions do occur they will be catastrophic with all the data from the cluster heads being lost in one collision, whereas it would only be the data lost from individual sensors in the single-hop case.

If the number of sensors, the Reliability and network Lifetime are multiplied together the following figures are obtained in the 12 cases: 705, 1371, 1868, 2329, 2755, 3200, 3164, 3408, 3785, 3592, 4215, and 4585. This gives a measure of how much data is received by the sink node, the Lifetime increases with the Density whilst the Reliability decreases. The figures indicate that the total number of packets of data received by the sink node will increase with the number of sensors, although data lost from sensors before they reach a cluster head is being ignored.

From the data one can observe that the Reliability, Lifetime and the number of sensors are related by the following formula:

\[
\frac{\text{Reliability}}{\text{Lifetime}} \times \text{Number of sensors} = \text{Constant} \tag{1}
\]

This equation states that the rate of successful packet reception per unit time is independent of the number of sensors. The constant in the equation above will depend on the parameters in the simulation and it is a reasonable model if the number of cluster heads is small. The number of sensors in the cluster is one way of measuring packets received, but again ignores data lost from sensors before it reaches the cluster.
head. Such lost data will increase with the number of sensors, so that the constant on the right hand side of the equation may have to be found for a small, medium or large number of sensors.

This experiment gives the following ‘constants’ in the 12 cases (to six decimal places):
0.000297, 0.000312, 0.000298, 0.000302, 0.000291, 0.000269, 0.000282, 0.000292, 0.000261, 0.000277, 0.000280, which because these figures are all very close to each other indicates that the model is reasonable.

Now, as there is a constant rate of successful packet reception per unit time, it follows from calculus that plotting the Reliability multiplied by the number of sensors against the Lifetime, a line $ax + b$ should be obtained, where $x$ represents the Lifetime and $a$ is ‘the’ constant obtained above.

The least-squares approximation method [116] was used to fit the line through the data points as illustrated in Figure 6, this yields the line:

$$0.00025x + 0.10993$$

(2)

Inverting this linear relationship the following equation is obtained:

$$\text{Lifetime} = 4000y - 439.72$$

(3)

where $y$ is the product of the Reliability with the number of sensors. The 4000 is the rate at which the Lifetime (energy) is used per unit of $y$. The constant 439.72 appears to be a measure of the Lifetime used in cluster head activities that are independent of the number of sensors.
Informal tests for radii less than 10m indicated that these would result in similar curves to those in Figure 6.1.

In Figure 6.3, the transmission radius for sensors and the target node have been increased from 10 to 15 metres (The radius choice matches the previous experiments.). All the nodes (include common sensors, sink nodes and target nodes) are located in the 10m x 10m square. The longest distance between a node and the target node is less than 14.15 metres. Thus, 15m radius is large enough for all nodes to receive the message from the target node. (The target transmits periodic signal to all the sensors.) The Lifetime for this experiment in Figure 6.3 (a) increases from 1421 to 2077, 2407, 2767, 3053, 3183, 3401, 3577, 3597, 3875, 3957 and finally to 4055. The Reliability in Figure 6.3 (b) decreased from 24.38% to 15.66%, 12.38%, 9.35%, 7.48%, 6.07%, 5.31%, 4.98%, 4.08%, 3.65%, 2.35% and finally to 1.87%. Comparing Figure 6.1 and Figure 6.3, this chapter concludes that the Lifetime and Reliability are not affected
much by the increase in Radius with the shape of the curves much the same as before. This will be further analyzed in the next section. However, there is a degree of experimental variation in the results as the cluster heads will be randomly deployed in different positions. So when the number of sensors is just two, the Radius should have no effect on the Reliability or Lifetime as the only transmissions these sensors have to make is as cluster heads.
Figure 6.3: (a) Number of Sensors, *Lifetime* Relationship. (b) Number of Sensors, *Reliability* Relationship.
6.4.2. **Radius—Lifetime—Reliability**

In order to analyze the effect of **Radius**, network **Lifetime** and **Reliability** on WSN performance, a series of experiments was carried out with a transmission **Radius** of each node starting at one metre, and then increasing in one-metre increments up to eight metres. The ability to communicate with the sink node for the sensors is limited by the transmission **Radius**. In this experiment, a fixed value of 20 sensors was set for the **Density**. The number of clusters for this application was two and the transmission **Radius** for the target node was also 15 metres (Having two cluster heads is the easiest situation to analyze from the viewpoint of data collisions at the sink node. 15 metre is large enough for the target node to transmit packets. In addition, the choice of this setting is the same as setting of the figures in the Appendix. ).

In Figure 6.4 (a) the value of the **Lifetime** starts at 3487 and then becomes 3427, 3431, 3475, 3495, 3567, 3423, and finally 3455.

In Figure 6.4 (b) the value of the **Reliability** starts at 4.79% and then becomes 4.75%, 4.67%, 4.96%, but at the fifth scenario it reaches its highest point of 5.09% and then drops to 5.08%, 5.03% and finally to 4.73%.
Figure 6.4: (a) Radius, *Lifetime* Relationship. (b) Radius, *Reliability* Relationship.
In Figure 6.4, the *Lifetime* is between 3400 and 3600 as the *Radius* increases from one to eight metres. On the other hand, the *Reliability* fluctuates between 4.6 and 5.2% as the *Radius* increases. The reason for this curve can be explained as follows: When the sensors located close to the sink node become cluster heads, they can transmit packets to the sink node. So the *Lifetime* and *Reliability* are mainly affected by the number of sensors that can communicate with the sink node. Thus, the effect of *Radius* seems very weak and the experiments indicate that the *Radius* parameter is not critical for LEACH.

### 6.4.3. Cluster—*Lifetime*—*Reliability*

The number of clusters is another key parameter for the network *Lifetime* and The number of clusters is another key parameter for the network *Lifetime* and *Reliability*. Figure 6.5 shows the effect of varying the number of clusters.

In Figure 6.5, as the number of clusters becomes larger, starting at 1, increasing to 2, 3, 5, 7 and finally to 10, the network *Lifetime* decreases but the *Reliability* reaches its highest value when the number of clusters equals seven. The transmission *Radius* for each sensor stays at 15 metres and the number of sensors is 20. The transmission *Radius* for the target node is also 15 metres. The result here for the *Lifetime* is as expected, with the number of sensors and *Radius* fixed. This work expects the *Lifetime* to decrease as the number of cluster heads increases, as these provide an increasing drain on the energy of the system.
Figure 6.5: (a) Number of Clusters, \textit{Lifetime} Relationship (20 Sensors). (b) Number of Clusters, \textit{Reliability} Relationship (20 Sensors).
Figure 6.6: Number of Clusters, *Lifetime* Relationship (30 Sensors). (b) Number of Clusters, *Reliability* Relationship (30 Sensors).
In Figure 6.6, as the number of clusters becomes larger, starting at 2, increasing to 5, 7, 10, 13 and finally to 15, the network Lifetime decreases but the Reliability reaches its highest value when the number of clusters equals 10. The transmission Radius for each sensor remains at 15 metres, but the number of sensors is now 30. The transmission Radius for the target node is again 15 metres.

In Figure 6.7, as the number of clusters becomes larger, starting at 2, increasing to 5, 8, 10, 15 and finally to 20, the network Lifetime decreases but the Reliability reaches its highest value when the number of clusters equals 10. The transmission Radius for each sensor and the target node is still 15 metres, but the number of sensors is now 40.

A large number of cluster heads will lose data through data collision at the sink node. This will become the dominant source of data loss at some point and so this work would expect the Reliability to decrease monotonically as the number of cluster heads increases after this point. Thus there must be some value for the number of cluster heads which gives the optimum Reliability, although this will in turn also be dependent on the number of sensors in the network. So researchers should make a trade-off between the network Lifetime and Reliability.
Figure 6.7: Number of Clusters, Lifetime Relationship (40 Sensors). (b) Number of Clusters, Reliability Relationship (40 Sensors).
6.5. *Lifetime Model*

This section will propose a simple relationship between the *Lifetime* and the number of sensors assuming all other parameters remain constant. The *Lifetime* can be modelled by a two parameter equation involving $E$, the energy used in cluster head activities per time unit and $e$, the energy used in normal sensor activities per time unit. (This thesis assumes all the sensors deployed have same energy consumption models)

The energy used in normal sensor activities should be independent of the number of sensors and therefore $e$ is constant. On the other hand $E$ will increase with the number of sensors as each cluster will contain more sensors and so its cluster head will have to aggregate (and transmit) more data. Nevertheless $E$ is assumed to be a constant by finding an average value for it.

Let $K$ be the total battery energy of a sensor, let $n$ be the total number of sensors and let $r$ be the number of clusters. Then the *Lifetime* should be governed by the equation

$$nK = rtE + (n - r)e$$  

(4)

where $t$ denotes the *Lifetime*. A cluster head acts as a normal sensor and also aggregates data from other sensors. It transmits all this aggregated data (including its own as a sensor) to the sink node. The cluster head acting as a normal sensor is unusual in that it does not have to transmit its data to a cluster head and for this reason the figure $E$ includes the energy used by the cluster head acting as a ‘normal’ sensor, rather than assigning it the value $e$ (which would be too high).

$$t = nK/[rE + (n - r)e]$$  

(5)
As \( n \to \infty, \ t \to K/e \), this of course is independent of \( r \) and \( E \). So the horizontal line \( t = K/e \) should form a horizontal asymptote in the Density-Lifetime graph. On the other hand when \( n = r, \ t = K/E \) as one would expect.

Figure 6.1 is comprised of the following twelve data points: (2, 1541), (4, 2097), (6, 2503), (9, 2779), (12, 3077), (15, 3307), (18, 3427), (21, 3475), (24, 3601), (30, 3707), (45, 3903), (60, 4043). The data for 15 and 30 sensors can be used to obtain an average value for \( E \).

Now

\[
15K = 6614E + 42991e \quad (6)
\]

and

\[
30K = 7414E + 103796e \quad (7)
\]

Eliminating \( K \) from these equations gives

\[
13228E + 85982e = 7414E + 103796e \quad (8)
\]

and so

\[
5814E = 17814e \quad (9)
\]

then

\[
e = 0.326372516E \quad (10)
\]
and so

\[ K = 1376.338722E \] \hspace{1cm} (11)

Using these approximations

\[ t = 1376.338722n/[2 + 0.326372516(n - 2)] \] \hspace{1cm} (12)

So as \( n \to \infty \), \( t \to 4217.079119 \), this should be an overestimation of the *Lifetime*, because of the average value taken for \( E \).

The graph of the function \((n, t)\) with \( t \) determined by the formula above (the red curve in Figure 6.8) has been overlaid with that of the *Lifetime* in Figure 6.1 (in blue in Figure 6.8), confirming that this is a good model.

![Figure 6.8: A Model of the Lifetime.](image)
If \( n \) is large compared to \( r \) a simpler relationship can be found by replacing \( n - r \) by \( n \) in the above equation, so that

\[
nK = rtE + nte
\]  

(13)

where \( t \) denotes the *Lifetime*.

Then

\[
t = \frac{nK}{[rE + ne]}
\]  

(14)

Repeating the above analysis (without assuming that \( r = 2 \)) yields

\[
e = 0.24225rE
\]  

(15)

and

\[
K = 1135.237983rE
\]  

(16)

Using these approximations

\[
t = 1135.237983n/[1 + 0.24225n]
\]  

(17)

independent of \( r \).

Of course in reality the (average) total battery energy of a sensor will be known, since this can be measured before deployment.
6.6. **Reliability Model**

The formula connecting the *Reliability* and *Lifetime* in Section 6.4 will now be married with the equation for the *Lifetime* found in Section 6.5 to obtain the following formula for the *Reliability*:

\[
Reliability = \frac{K}{rE+(n-r)e} \times \text{Constant}
\]  \hspace{1cm} (18)

where the constant has to be determined by experimentation as shown in Section 6.4. Based on the data in Figure 6.1 \((r = 2)\) and the arithmetic mean 0.000288 of the constants found in Section 6.5, the following equation is obtained:

\[
Reliability = \frac{0.396385551}{2+0.326372516(n-2)}
\]  \hspace{1cm} (19)
Plotting this graph (in red in Figure 6.9) over that of Figure 6.2 for the Reliability (in blue in Figure 6.9) shows that this is a good model for the Reliability.

Figure 6.9: A Model of the Reliability.

This chapter has detailed the basic version of LEACH, which is still widely used. The LEACH protocol has been successfully modelled mathematically. However, over the years new protocols based on LEACH have been developed to obtain improved results from the WSN. Quadrature-LEACH (detailed in Section 3.2.4) is one of these new protocols. In Q-LEACH, network is partitioned into sub-sectors. The clusters formed in these sub-sectors become more deterministic. In the simulation of the literature [100], the Lifetime of Q-LEACH can be improved from 1700 to 2900. So the following prediction can be proposed based on above results.
\[ \text{Lifetime of } Q - \text{LEACH} = 1135.237983n / [1 + 0.24225n] \times 2900 / 1700 \]  \hspace{1cm} (20)

and so,

\[ \text{Lifetime of } Q - \text{LEACH} = 1936.582442n / [1 + 0.24225n] \]  \hspace{1cm} (21)

6.7. Summary

In this chapter, a series of experiments with the J-Sim simulation tool was conducted and several conclusions were reached:

(a) The network Lifetime increases as the number of sensors increases.

(b) There is a linear relationship between the number of packets received by the sink node and the Lifetime.

(c) The effect of Radius is weak indicating that this parameter is not critical for LEACH.

(d) The number of clusters affects data collision and energy and consequently the Reliability and network Lifetime.

Based on these simulation results, some models in the form of equations have been constructed.

After an analysis of the significant hierarchical routing protocol LEACH, the next chapter will focus on one of the typical location-based routing protocols, the Nearest Closer protocol, and will provide some J-Sim simulation results and evaluation models for it.
Chapter 7: Evaluation Models for the Nearest Closer Protocol

Generally multi-hop routing in a wireless sensor network is more efficient than that of single-hop as discussed in Section 5.1, particularly if the network is large. In multi-hop routing data is transmitted from sensor to sensor, using some protocol, until it reaches the sink node.

As communication becomes more and more complex using a multi-hop routing protocol, data collision will be more complex than for a single-hop protocol (Data collision occurs at common sensors in the Nearest Closer (NC) protocol, but it just occurs at the sink node in Single-hop), and this in turn affects the Reliability and Lifetime. In order to decrease the effect of data collision in wireless sensor networks the transmission radius should be set appropriately. The metrics used in simulations normally reflect the goal of the designed algorithm. The optimal routing path to transmit the collected data from a source node to a sink can be defined by the metric. Most routing schemes simply use hop count as the metric [117], where hop count is the number of transmissions on a route from a source to a destination. However, if nodes can adjust their transmission power (knowing the location of their neighbours) the constant per hop metric can be replaced by a power metric that depends on the distance between nodes.

The Nearest Closer protocol (detailed in Section 3.3.3) is both a typical location-based routing protocol and also a multi-hop routing protocol. Consequently this work has implemented the protocol in J-Sim as an example. To implement this protocol each node has to know its own position, the position of its neighbours within its transmission range, and the position of the sink node. The main idea in the Nearest
Closer protocol is that the transmitter sensor will transmit to its nearest neighbour that is closer to the sink node.

### 7.1. System Architecture and Experimental Set-up

Slotted ALOHA protocol (detailed in 2.1.3) has been selected as the MAC layer protocol. Slotted ALOHA is a type of TDMA transmission system and it improves contention management through the use of beaconing. Slotted ALOHA can make a single active sensor nearly continuously transmit at full channel rate, thus better results can be obtained for the Nearest Closer protocol.

The simulated area for the following experiments is defined as a 10 metre by 10 metre square with randomly deployed nodes. The routing protocol is the Nearest Closer. The sink node for this application is located in the centre of this area.

### 7.2. Results

#### 7.2.1. Density—Lifetime—Reliability

A series of experiments was carried out with the number of sensors starting at 10, and increasing in increments of 10 up to 300 sensors. The transmission radius for each sensor was fixed at 15 metres as this is large enough to transmit data anywhere within the square. The Reliability and Lifetime for this experiment are tabulated in Table 7.1.
In Figure 7.1, the relationship between the number of sensors and the *Reliability* is very clear. The *Reliability* for this application increased as the number of sensors increased to 40, when it reached its highest value. It then essentially decreased as the number of sensors increased from 40 to 300, when it reached its lowest value. This may be explained by observing that as the density of nodes increases, more sensors will join the data transmission process and consequently communication among the sensors will become more and more complex. So dropped data due to data collision and latency cannot be ignored. Consequently it is reasonable to expect the *Reliability* to decrease with the *Density*. 

Figure 7.1: Density, Reliability Relationship for NC.
In Figure 7.2, the \textit{Lifetime} reached its lowest value when the number of sensors equalled 220, whereas the highest value of the \textit{Lifetime} occurred when there were 10 sensors.

In the NC protocol, sending a packet to the sink node will require each sensor to transmit its data down transmission trees to the nearest sensors to the sink node. These nearest sensors (only about two or three) will be receiving all the data in the network and transmitting it all to the sink node, using a large amount of energy. Thus a NC network will generally die fairly rapidly from the sink node outwards. If the LEACH network with two or three clusters was run with fixed cluster heads then one would expect that its \textit{Lifetime} would be very short with each cluster head having to collect and transmit all the data in the network (like a NC network at outset). However, the
crucial feature of LEACH is that the cluster heads are rotated amongst the sensors, allowing this energy burden to be spread evenly amongst the sensors and so leading to a much longer *Lifetime*.

In Figure 7.2, note that when the number of sensors equals 40, the *Reliability* was 93.64%, its highest level, but with this number the *Lifetime* is fairly short. This illustrates that it is possible for users to choose an optimum *Density* value for this application depending on the *Reliability* and *Lifetime* required. Section 7.3 will build a *Lifetime* model and explain the reason why the *Lifetime* decreases as the number of sensors increases to 130.
Table 7.1: Number of Sensors, Reliability, Lifetime for NC.

<table>
<thead>
<tr>
<th>No. of sensors</th>
<th>Reliability %</th>
<th>Lifetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>89.08</td>
<td>702</td>
</tr>
<tr>
<td>20</td>
<td>92.53</td>
<td>581</td>
</tr>
<tr>
<td>30</td>
<td>93.21</td>
<td>530</td>
</tr>
<tr>
<td>40</td>
<td>93.64</td>
<td>438</td>
</tr>
<tr>
<td>50</td>
<td>91.81</td>
<td>341</td>
</tr>
<tr>
<td>60</td>
<td>90.63</td>
<td>324</td>
</tr>
<tr>
<td>70</td>
<td>89.72</td>
<td>282</td>
</tr>
<tr>
<td>80</td>
<td>90.03</td>
<td>280</td>
</tr>
<tr>
<td>90</td>
<td>89.78</td>
<td>277</td>
</tr>
<tr>
<td>100</td>
<td>89.05</td>
<td>274</td>
</tr>
<tr>
<td>110</td>
<td>88.98</td>
<td>258</td>
</tr>
<tr>
<td>120</td>
<td>89.86</td>
<td>230</td>
</tr>
<tr>
<td>130</td>
<td>88.83</td>
<td>228</td>
</tr>
<tr>
<td>140</td>
<td>88.67</td>
<td>244</td>
</tr>
<tr>
<td>150</td>
<td>88.61</td>
<td>239</td>
</tr>
<tr>
<td>160</td>
<td>88.28</td>
<td>234</td>
</tr>
<tr>
<td>170</td>
<td>88.07</td>
<td>224</td>
</tr>
<tr>
<td>180</td>
<td>87.97</td>
<td>227</td>
</tr>
<tr>
<td>190</td>
<td>87.31</td>
<td>232</td>
</tr>
<tr>
<td>200</td>
<td>86.31</td>
<td>233</td>
</tr>
<tr>
<td>210</td>
<td>84.99</td>
<td>228</td>
</tr>
<tr>
<td>220</td>
<td>83.91</td>
<td>223</td>
</tr>
<tr>
<td>230</td>
<td>83.51</td>
<td>226</td>
</tr>
<tr>
<td>240</td>
<td>83.24</td>
<td>249</td>
</tr>
<tr>
<td>250</td>
<td>83.02</td>
<td>248</td>
</tr>
<tr>
<td>260</td>
<td>81.46</td>
<td>246</td>
</tr>
<tr>
<td>270</td>
<td>79.04</td>
<td>251</td>
</tr>
<tr>
<td>280</td>
<td>77.15</td>
<td>250</td>
</tr>
<tr>
<td>290</td>
<td>76.84</td>
<td>240</td>
</tr>
<tr>
<td>300</td>
<td>75.89</td>
<td>248</td>
</tr>
</tbody>
</table>
7.2.2. *Radius—Lifetime—Reliability*

A series of experiments were carried out with the transmission radius starting at 1 metre and increasing by one-metre increments to 10 metres. The number of sensors was fixed at 200.

Figure 7.3 shows the relationship between the *Reliability* and the transmission radius. It is necessary that all the sensor nodes adopt the Nearest Closer transmission technique to deliver messages. When the radius equals one metre, the *Reliability* is at its lowest; however, if the radius is greater than one metre, the value of Reliability remains relatively stable, varying by less than 10%. The reason for this could be explained as follows:
For a one-metre radius connectivity, a lot of data will be lost in the transmission process, and hence the Reliability in the system is adversely affected. On the other hand, as the radius increases more data collisions will occur, so that the Reliability will be affected as well.

The value of the Reliability fluctuates between 85 and 90% for a radius greater or equal to two metres.

Figure 7.4 shows the relationship between the Lifetime and the transmission radius. There is no clear trend in this figure, with the Lifetime varying only between 236 and 254 and it may be considered constant for any radius between 1 and 10 metres. If the number of sensors is adjusted to 20, the results are as follows:
Figure 7.5 shows the relationship between the Reliability and the transmission radius. The Reliability for this application increased as the transmission radius increased from 1 to 5 metres. The Reliability reached 93.28% when the radius equalled 5 metres. As the radius increased from 5 to 10 metres, the Reliability value fluctuated slightly since the radius is sufficiently large to transmit the data for this Nearest Closer protocol to the sink node.
Figure 7.6: Radius, Lifetime Relationship for NC (20 Sensors).

Figure 7.6 shows the relationship between the Lifetime and the transmission radius. As the radius increased from 1 to 10 metres in one-metre increments, the Lifetime decreased from 757 to 569. The reason for this drop could be due to the fact that as the radius increased, communication among the sensors became more complex, thus much more energy was consumed leading to a decreasing Lifetime.

7.3. Lifetime Model

Hop count is one way to measure the energy requirement of a routing task, thus using a constant metric per hop. However, if nodes can adjust their transmission power (knowing the location of their neighbours) the constant metric can be replaced by a power metric \( u(d) = e + d^\alpha \) (or some variation of this) for some constants \( \alpha \) and \( e \) that depend on the distance \( d \) between nodes. The value of \( e \), which includes energy loss due to start up, collisions, retransmissions, and acknowledgements, is
relatively significant, and protocols using any kind of periodic hello messages are extremely energy inefficient.

The basic idea for the NC protocol model to be constructed in this section is that the energy consumed by an average sensor in sending all its data to its nearest neighbour is of the form $w(s)E(d^2)$, where $E(d^2)$ is the expected value of the square of the distance between sensors and $w(s)$ is the average number (weight) of packets of data to be sent to its nearest neighbour. This figure also represents the average amount of energy to send one packet of data to the sink node.

In the NC protocol one possible parameter that could affect the *Lifetime* is the number of paths or trees formed by the sensors. This is not the case as demonstrated by the following experiments, which are of independent interest. Using J-Sim it is difficult to extract the paths that sensors transmit along to the sink node. For this reason a program was written in the algebraic software package Magma [118] to not only compute the individual paths, but to marry them together in order to form the initial distinct trees. Of course during the *Lifetime* of the network various sensors will die, normally from the centre outwards, and new trees will be formed (The sink node is in the middle. Tree roots are close to the sink node and they will die quickly as they do receive and transmit data frequently. In this thesis, all the common sensors are randomly deployed and have the same power.). The mean number of trees corresponding to different numbers of sensors is given in Table 7.2. This mean number of trees was found by running the program 1,000 times for each given number of sensors. The effect of randomly placed sensors in each experiment can be ignored as the value was averaged from the results of 1000 experiments. It can be seen from this table that the number of trees is almost independent of the number of sensors (excluding very small values) and is of the order of 2.55 trees.
Table 7.2: Mean Number of Trees.

<table>
<thead>
<tr>
<th>No. of Sensors</th>
<th>Trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2.537</td>
</tr>
<tr>
<td>20</td>
<td>2.527</td>
</tr>
<tr>
<td>30</td>
<td>2.561</td>
</tr>
<tr>
<td>40</td>
<td>2.559</td>
</tr>
<tr>
<td>50</td>
<td>2.531</td>
</tr>
<tr>
<td>60</td>
<td>2.550</td>
</tr>
<tr>
<td>70</td>
<td>2.542</td>
</tr>
<tr>
<td>80</td>
<td>2.534</td>
</tr>
<tr>
<td>90</td>
<td>2.553</td>
</tr>
<tr>
<td>100</td>
<td>2.549</td>
</tr>
<tr>
<td>110</td>
<td>2.568</td>
</tr>
<tr>
<td>120</td>
<td>2.578</td>
</tr>
<tr>
<td>130</td>
<td>2.555</td>
</tr>
<tr>
<td>140</td>
<td>2.543</td>
</tr>
<tr>
<td>150</td>
<td>2.574</td>
</tr>
<tr>
<td>160</td>
<td>2.550</td>
</tr>
<tr>
<td>170</td>
<td>2.557</td>
</tr>
<tr>
<td>180</td>
<td>2.552</td>
</tr>
<tr>
<td>190</td>
<td>2.575</td>
</tr>
<tr>
<td>200</td>
<td>2.582</td>
</tr>
<tr>
<td>210</td>
<td>2.574</td>
</tr>
<tr>
<td>220</td>
<td>2.600</td>
</tr>
<tr>
<td>230</td>
<td>2.571</td>
</tr>
<tr>
<td>240</td>
<td>2.573</td>
</tr>
<tr>
<td>250</td>
<td>2.569</td>
</tr>
<tr>
<td>260</td>
<td>2.551</td>
</tr>
<tr>
<td>270</td>
<td>2.575</td>
</tr>
<tr>
<td>280</td>
<td>2.543</td>
</tr>
<tr>
<td>290</td>
<td>2.563</td>
</tr>
<tr>
<td>300</td>
<td>2.562</td>
</tr>
</tbody>
</table>
The conclusion from Table 7.2 is that the mean number of trees is independent of the number of sensors and therefore can be discounted as having any effect on the Reliability or Lifetime. The reason that on average two or three trees occur can be satisfactorily explained by considering the diagram below.

![Typical Tree Zones](image)

**Figure 7.7: Typical Tree Zones.**

In Figure 7.7 the points $A$, $B$ and $C$ represent the three nearest sensors to the sink node (which is in the centre of the square), amongst the randomly distributed sensors in the region. The perpendicular bisectors of the line segments joining $A$, $B$ and $C$ to the sink node have then been constructed. It is highly probable that the next furthest sensor will be in one of the three zones containing $A$, $B$ or $C$ and its corresponding perpendicular bisector. Thus, in this diagram three trees will probably result with $A$, $B$ and $C$ as their roots. Then, in ‘zone’ $A$ it may be the case that further out in the tree sensors cross over into zone $B$ or $C$. It is fairly easy to draw diagrams where two or four trees will result, but only a very lop-sided distribution will result in just one tree if a reasonable number of sensors is being distributed.
To verify this theoretical analysis a detailed study of three cases was undertaken (The cases were chosen using a 'small', 'medium' and 'large' number of sensors from our experiments, which range from 10 to 300 sensors. The choices of 10 and 300 illustrate the different tree formations when the numbers of sensors are sparsely or fairly densely packed in the square and the choice of 40 illustrates what happens when the number of sensors are moving from being sparsely packed to less so. The choices of 10 and 40 also allow the individual tree distributions to be listed (only for two trees for 40 sensors); whereas for 300 sensors the number of possible tree distributions is also large and so when a distribution occurs it tends to occur with frequency 1 in the 100 experiments.):

For 10 sensors the number of trees and their frequency in a trial of 100 experiments were one with frequency 2, two with frequency 42, three with frequency 51 and four with frequency 5. To illustrate the type of distributions that can occur in this limited case the number of sensors per tree is recorded in Table 7.3 together with the frequency of each type. The entry (9, 1), 3 means that there were two trees, nine sensors were in the first and one in the second, moreover this distribution occurred three times in the 100 trials.
Table 7.3: Tree Distributions for 10 Sensors.

<table>
<thead>
<tr>
<th>Distribution type</th>
<th>(9, 1)</th>
<th>(8, 2)</th>
<th>(7, 3)</th>
<th>(6, 4)</th>
<th>(5, 5)</th>
<th>(8, 1, 1)</th>
<th>(7, 2, 1)</th>
<th>(6, 3, 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>3</td>
<td>9</td>
<td>9</td>
<td>14</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution type</th>
<th>(6, 2, 2)</th>
<th>(5, 4, 1)</th>
<th>(5, 3, 2)</th>
<th>(4, 4, 2)</th>
<th>(4, 3, 3)</th>
<th>(5, 3, 1, 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>2</td>
<td>5</td>
<td>19</td>
<td>5</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution type</th>
<th>(5, 2, 2, 1)</th>
<th>(4, 3, 2, 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

For 40 sensors the number of trees with frequencies was one with frequency 3, two with frequency 48, three with frequency 44 and four with frequency 5. This thesis will only list the distributions for two trees (There would be 44 three-tree distributions to list and 5 four-tree distributions, the number of potential three-tree distributions (it is the number of partitions of 40 into three numbers) that can arise for 40 sensors is large and when such a distribution does occur its frequency is generally one, listing these would provide little information. Some statistical analysis on the number of sensors in the two-tree cases will be carried out on the next page of this thesis, and such analysis is much harder to perform with three or more trees. The number of trees and their frequencies are the most important outcomes listed here for the 10, 40 and 300 sensor experiments, as Table 7.2 illustrates the average number of trees is surprisingly independent of the number of sensors (for 10 or more sensors).):
Table 7.4: Two Tree Distributions for 40 Sensors.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>(39, 1)</th>
<th>(38, 2)</th>
<th>(36, 4)</th>
<th>(34, 6)</th>
<th>(33, 7)</th>
<th>(32, 8)</th>
<th>(31, 9)</th>
<th>(30, 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution</th>
<th>(29, 11)</th>
<th>(28, 12)</th>
<th>(27, 13)</th>
<th>(26, 14)</th>
<th>(25, 15)</th>
<th>(24, 16)</th>
<th>(23, 17)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution</th>
<th>(22, 18)</th>
<th>(21, 19)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

For 300 sensors the number of trees with frequencies was one with frequency 1, two with frequency 42, three with frequency 53 and four with frequency 3. The distributions for two trees took various values from (299, 1) to (152, 148). For the cases of two trees the average length of the two trees for 10 sensors was 6.69 and 3.31 with a sample standard deviation of 1.199. The ratio between these average lengths is 2.02. For 40 sensors the corresponding figures are 26.667 and 13.333 with a sample standard deviation of 4.646. The ratio between these average lengths is 2. For 300 sensors the corresponding figures are 213.929 and 86.071 with a sample standard deviation of 39.38. The ratio between these average lengths is 2.49.

The reason for using the average tree length as a test statistic is that it can be generalized to more than two trees. In the latter part of this chapter it is assumed that the sensors are equally distributed amongst the trees. The results above indicate in the two-tree case that the ratio between the average lengths of the trees is about 2 or 2.5 to
1. The reason for this may be that the shape of the square influences the sensors into one large and one small tree, this merits further investigation from a mathematical or statistical perspective.

To illustrate the shape of trees that can arise under the NC protocol some specific (and not necessarily typical) examples are given in the six figures below for 10 and 40 sensors. In each of these diagrams the small circle in the middle stands for the sink node and the individual trees are differentiated by colour.

Figure 7.8: 10 Sensors, One Tree.
Figure 7.9: 10 Sensors, Two Trees.

Figure 7.10: 10 Sensors, Three Trees.
Figure 7.11: 40 Sensors, Two Trees.

Figure 7.12: 40 Sensors, Three Trees.
For the NC protocol the relation ‘is the closest neighbour to’ is not symmetric, that is, $B$ can be the closest neighbour to $A$ without $A$ being the closest neighbour to $B$. Thus for this protocol it is necessary to find the closest neighbour to a sensor $A$, but with the stipulation that the neighbour is closer to the sink node than $A$. It is also the case that the sink node could be the closest neighbour to $A$, but the sink node is a fixed point and therefore does not fit in with the assumption that this work is dealing with a random distribution of sensors.

The expected distance $d$ between a sensor node and its closest neighbor, which is closer to the sink node, is given in the second column of Table 7.5 and the expected square of the distance is given in the third column. Finally, in the fourth column, the average weight of the sensors is given where each branch in the various trees count as weight one. This is a measure of the number of packets of data an average sensor has
to transmit to its nearest neighbor, and it is also the average number of hops from the sensors to the sink node. The results of these simulations are tabulated in Table 7.5 (In Table 7.5, \(E(d)\) is the expected value of the distance between sensors, \(E(d^2)\) is the expected value of the square of the distance between sensors and \(w(s)\) is the average number (weight) of packets of data to be sent to its nearest neighbour.)

It should be noted for those sensors whose nearest neighbour is the sink node that the distribution of the distance and distance squared to the sink node is slightly different to those between sensors, and the averages are always slightly higher than those given in Table 7.5, with the exception of the 10 sensor case when there is a significant difference.

Two results emanate from this table. The first is that \(w(s)E(d)\) is nearly constant as it only ranges from 5.16 to 5.45 and is a measure of the average distance from a sensor to the sink node measured along the hop path. The second is that \(w(s)E(d^2)\) decreases as the number of sensors goes up, this shows that the energy expended by a sensor in receiving and sending received packets to its nearest neighbour decreases as the number of sensors increases. It also represents a measure of the average amount of energy needed to transmit one packet of data to the sink node. Two theories are proposed to explain why the Lifetime decreases as the number of sensors goes up.
Table 7.5: Expected Distances and Weights, Variable Sensors.

<table>
<thead>
<tr>
<th>Sensors</th>
<th>$E(d)$</th>
<th>$E(d^2)$</th>
<th>$w(s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2.1945</td>
<td>6.0533</td>
<td>2.3526</td>
</tr>
<tr>
<td>20</td>
<td>1.5985</td>
<td>3.2761</td>
<td>3.3421</td>
</tr>
<tr>
<td>30</td>
<td>1.3223</td>
<td>2.2444</td>
<td>4.0821</td>
</tr>
<tr>
<td>40</td>
<td>1.1512</td>
<td>1.7023</td>
<td>4.7152</td>
</tr>
<tr>
<td>50</td>
<td>1.0264</td>
<td>1.3535</td>
<td>5.2218</td>
</tr>
<tr>
<td>60</td>
<td>0.9402</td>
<td>1.1350</td>
<td>5.7941</td>
</tr>
<tr>
<td>70</td>
<td>0.8672</td>
<td>0.9679</td>
<td>6.2249</td>
</tr>
<tr>
<td>80</td>
<td>0.8104</td>
<td>0.8450</td>
<td>6.6531</td>
</tr>
<tr>
<td>90</td>
<td>0.7657</td>
<td>0.7545</td>
<td>7.0685</td>
</tr>
<tr>
<td>100</td>
<td>0.7256</td>
<td>0.6759</td>
<td>7.4110</td>
</tr>
<tr>
<td>110</td>
<td>0.6926</td>
<td>0.6155</td>
<td>7.7859</td>
</tr>
<tr>
<td>120</td>
<td>0.6618</td>
<td>0.5634</td>
<td>8.1589</td>
</tr>
<tr>
<td>130</td>
<td>0.6360</td>
<td>0.5200</td>
<td>8.4701</td>
</tr>
<tr>
<td>140</td>
<td>0.6124</td>
<td>0.4817</td>
<td>8.7956</td>
</tr>
<tr>
<td>150</td>
<td>0.5910</td>
<td>0.4485</td>
<td>9.0999</td>
</tr>
<tr>
<td>160</td>
<td>0.5724</td>
<td>0.4210</td>
<td>9.4502</td>
</tr>
<tr>
<td>170</td>
<td>0.5553</td>
<td>0.3957</td>
<td>9.7218</td>
</tr>
<tr>
<td>180</td>
<td>0.5394</td>
<td>0.3734</td>
<td>9.9967</td>
</tr>
<tr>
<td>190</td>
<td>0.5240</td>
<td>0.3525</td>
<td>10.3059</td>
</tr>
<tr>
<td>200</td>
<td>0.5108</td>
<td>0.3346</td>
<td>10.5684</td>
</tr>
<tr>
<td>210</td>
<td>0.4983</td>
<td>0.3183</td>
<td>10.7823</td>
</tr>
<tr>
<td>220</td>
<td>0.4874</td>
<td>0.3046</td>
<td>11.0709</td>
</tr>
<tr>
<td>230</td>
<td>0.4759</td>
<td>0.2904</td>
<td>11.2847</td>
</tr>
<tr>
<td>240</td>
<td>0.4658</td>
<td>0.2785</td>
<td>11.5108</td>
</tr>
<tr>
<td>250</td>
<td>0.4562</td>
<td>0.2670</td>
<td>11.8269</td>
</tr>
<tr>
<td>260</td>
<td>0.4466</td>
<td>0.2559</td>
<td>12.0056</td>
</tr>
<tr>
<td>270</td>
<td>0.4393</td>
<td>0.2475</td>
<td>12.2536</td>
</tr>
<tr>
<td>280</td>
<td>0.4308</td>
<td>0.2378</td>
<td>12.4573</td>
</tr>
<tr>
<td>290</td>
<td>0.4231</td>
<td>0.2296</td>
<td>12.7054</td>
</tr>
<tr>
<td>300</td>
<td>0.4160</td>
<td>0.2216</td>
<td>12.9399</td>
</tr>
</tbody>
</table>
### 7.3.1. Early Termination

To explain early termination it is useful to consider the situation when two trees occur. These two tree roots have to receive and transmit all the sensor data to the sink node and therefore have long transmission times with one time slot per packet. A transmission break, which in the definition of the Lifetime is interpreted as the end of the Lifetime, could result when both of the root sensors do not possess sufficient energy to transmit their data to the sink node. In a sense this is like the situation for the LEACH protocol with the root sensors acting as fixed cluster heads and the Lifetime being controlled by the lifetimes of those sensors. A transmission break could also result if there is a catastrophic data collision in which all the packets of data from the root sensors collide with each other at the sink node, and from which the system cannot recover. It is highly probable in a large system that some data collision will occur at the sink node.

### 7.3.2. Data Build-Up

As explained above, the tree root sensors have long transmission times to transmit their data in a number of slots. Each sensor can transmit and receive data, but cannot do both simultaneously, also in the slotted ALOHA MAC design transmissions in a tree are sequenced. So the sensors immediately prior to a root sensor will receive more data than previously when the root sensor transmits, which will subsequently be transmitted to the root sensor and then the pattern will be repeated. The situation will become more complicated higher up the tree, but generally the amount of data to be transmitted by a sensor will increase with time in a non-linear manner. In particular the root sensors will have more and more packets of data to transmit as data builds up in the tree. So for a fixed number of sensors the amount of energy required for a root sensor to transmit its packets of data to the sink node will increase with time. Delays in the slotted ALOHA protocol will lead to data build-up and data collision will result in retransmissions.
If a varying number of sensors is considered, then the amount of energy needed to send a frame of packets may increase with the number of sensors despite the fact that \( w(s)E(d^2) \) decreases with the number of sensors. The energy required to transmit such a frame of data (in reality such a frame may not be transmitted in consecutive slots) will increase with the number of sensors if the frame length (the number of packets) exceeds the ratio between \( E(d^2) \) for the corresponding number of sensors; this latter ratio is always less than 1.85 between successive values in Table 7.5 and approaches 1 as the number of sensors increases, so that it is reasonable to expect that the corresponding ratio of frame lengths will exceed this. This effect is balanced by the fact that in a sparser system, sensors will transmit and receive data more often than in a denser system. It is also of note that most energy is used in data transmission rather than reception; however, energy used for data reception could be significant when compared to energy use in a sparse system (The denser and sparser system is measured by the number of sensors in the same area).

The conclusion is that \( w(s) \) may not be the correct weight to be attached to an average sensor, but instead represents a lower bound for the weight.

The two theories can be related, in that with data build-up the root sensors may end up transmitting a large amount of data to the sink node using a lot of energy in the process. This could ultimately lead to root sensors having insufficient energy to transmit the frame or a catastrophic data collision, in either case this will lead to early termination of the system. In reality both theories proposed here contribute to the \textit{Lifetime} figures obtained with the latter theory difficult to quantify in a model, because of the complexity of the trees.

Nevertheless, the \textit{Lifetime} in general can be modelled by only considering the roots of the normally two or three trees. The case of just one tree occurring is of interest for a small number of sensors, for in this scenario the system should run until the only root
sensor dies naturally, at that time the system might interpret the death as a transmission break or the next sensor(s) out will take over the role of the root(s). In either case the *Lifetime* should be much longer for just one tree. The basic model constructed here is of the form

\[ \frac{E(T)K}{nE(d^2)} \]  

(1)

where \( n \) is the number of sensors, \( K \) is the total initial energy of a sensor and \( E(T) \) is the expected number of trees. Here \( n/E(T) \) represents the average weight per tree and since \( E(T) \) is nearly constant, the model should roughly resemble the curve \( 1/n \). The model in particular assumes that the sensors are evenly distributed amongst the trees and disregards data build-up, it should therefore represent an upper bound for the *Lifetime* once the system is running at full capacity. This would imply that the *Lifetime* decreases monotonically as the number of sensors increases; however, this basic model assumes that the system runs at full capacity from the start and ignores idle slots in the slotted ALOHA design. Thus there is a base value below which the *Lifetime* will not fall, as the system will behave in a manner akin to a sparser system during its initial phase and there will furthermore always be a number of idle slots. This base value will be taken to be the smallest *Lifetime* value of 223 that occurs when \( n = 220 \). So the revised model is:

\[ (1 - \lambda(n))223 + \lambda(n) \frac{E(T)K}{nE(d^2)} \]  

(2)

where \( \lambda(n) \) is a proportionality measure with \( \lambda(220) = 0 \). Now from the *Lifetime* data \( \lambda(n) \) will be very close to 0 for all \( n \geq 120 \). Setting \( \lambda(10) = 1 \) yields the value \( K = 16750 \). Computation gives the following values for \( \lambda(n) \):
Table 7.6: Evaluation of the Proportionality Measure.

<table>
<thead>
<tr>
<th>n</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>λ(n)</td>
<td>1</td>
<td>0.846</td>
<td>0.741</td>
<td>0.529</td>
<td>0.292</td>
<td>0.250</td>
<td>0.146</td>
<td>0.141</td>
<td>0.133</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n</th>
<th>100</th>
<th>110</th>
<th>120</th>
<th>130</th>
<th>140</th>
<th>150</th>
<th>160</th>
<th>170</th>
<th>180</th>
<th>190</th>
</tr>
</thead>
<tbody>
<tr>
<td>λ(n)</td>
<td>0.125</td>
<td>0.085</td>
<td>0.017</td>
<td>0.012</td>
<td>0.051</td>
<td>0.027</td>
<td>0.002</td>
<td>0.009</td>
<td>0.021</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n</th>
<th>200</th>
<th>210</th>
<th>220</th>
<th>230</th>
<th>240</th>
<th>250</th>
<th>260</th>
<th>270</th>
<th>280</th>
<th>290</th>
<th>300</th>
</tr>
</thead>
<tbody>
<tr>
<td>λ(n)</td>
<td>0.024</td>
<td>0.012</td>
<td>0</td>
<td>0.007</td>
<td>0.062</td>
<td>0.059</td>
<td>0.064</td>
<td>0.065</td>
<td>0.040</td>
<td>0.059</td>
<td></td>
</tr>
</tbody>
</table>

The parameter $\lambda(x)$ can be piecewise approximated by

$$1.1438571428571 - 0.01501071428514x \quad (3)$$

for $10 \leq x \leq 70$, and

$$0.3954 - 0.0029514285714286x \quad (4)$$

for $80 \leq x \leq 130$ by fitting straight lines through the given values. An alternative that will yield better local approximations is to interpolate between the known values of $\lambda$. Thus if $n \leq x < n + 10$, then

$$\lambda(x) \equiv \left(1 - \frac{x - n}{10}\right)\lambda(n) + \frac{x - n}{10}\lambda(n + 10) \quad (5)$$

The general equation obtained for $10 \leq x \leq 133$ is
and the value of \( E(d^2) \) for \( x \) sensors can again either be interpolated between the known values in Table 7.5 or by using the fitted formula

\[
(1 - \lambda(x))223 + \lambda(x)\frac{42712.5}{xE(d^2)}
\]

(6)

for \( 10 \leq x \leq 300 \). Using the latter formula (which is a good fit) yields at worst an implicit equation for the Lifetime for \( 10 \leq x \leq 130 \). Finally, it could be stated that the Lifetime is essentially constant for \( x \geq 140 \) with a value between 223 and 250 depending on the topology of the sensors.
Figure 7.14: (a) Lifetime Model (Red), Lifetime (Blue), 10 – 70 Sensors; (b) Lifetime Model (Red), Lifetime (Blue), 80 – 130 Sensors.
In both Figure 7.14 (a) and (b) the Lifetime model is nearly a straight line and is clearly a better fit for \( \leq 70 \) sensors.

### 7.5. Reliability Model

Using the NC protocol three principal causes for data collision can be identified:

(a) Out of range or connectivity: A sensor may have insufficient transmission radius to reach its nearest neighbour at some stage during the lifetime, and from that time all its data (including data that it receives from other sensors) will not reach the sink node. This type of data loss does not occur in the 10 by 10 metre simulated area considered in this thesis, provided the transmission radius is \( \geq 5\sqrt{2} \) metres.

(b) Sensor complexity: Two or more sensors sending data to the same nearest neighbour could have their signals collide at the neighbour if they use the same slot or slots. The larger the average weight of a sensor (with the weight being a measure of the packets to be transmitted) the more chance there will be of such a data collision, since the number of transmission slots increases accordingly. Table 7.5 shows that the average weight increases with the number of sensors, and that data build-up will also increase the probability of data collision.

(c) Data collision at the sink node: Different sensors may transmit packets to the sink node at the same time, so data collision at the sink node cannot be ignored. From Table 7.2 the number of trees for this application is around 2.5. In other words, the number of sensors in this application doesn’t affect the number of trees very much. The larger the number of sensors, the longer the time required to transmit data from sensors to the sink node. So the sensors will contact the sink node more frequently. The smaller the number of sensors,
the more data could be sent to the sink node in one application. Thus, data collision at the sink node is more complex.

The Reliability obtained in the experiments measures the data loss that arises from out of range sensors, sensor complexity and data collision at the sink node. For a large radius there are no out of range sensors, but then it is a question of which out of complexity and data collision at the sink node is the dominant cause of data loss, with the former increasing and the latter decreasing as the number of sensors increases.

When the transmission radius is 15 metres the conclusion from Figure 7.2 is that data collision at the sink node is the dominant factor for less than or equal to 40 sensors and from then on complexity is the dominant factor. This would then explain why the Reliability increases as the number of sensors increases to 40 sensors, and then decreases as the number of sensors increases beyond 40.

There is a known result for the slotted ALOHA design using the Single-hop protocol that is relevant here. This measures the efficiency of the system, that is the long run fraction of successful slots when there are \( n \) (for \( n \) large) sensors each with many packets to send. The efficiency is \( q(1 - q)^{n-1} \), where each sensor transmits a packet in a slot with probability \( q \). This result will apply at each receiving sensor (and the sink node) individually in the NC protocol although the result will depend on the number of sensors transmitting to the receiving sensor in question. However, trying to marry these results together for the whole system under the NC protocol would involve a detailed analysis of the trees and is certainly impractical to analyze.

In the following table the Reliability (as a percentage) divided by the Lifetime is listed for the 15-metre transmission radius case, which gives (100 times) the Reliability per unit Lifetime \( R(n)/L(n) \):
Table 7.7: Evaluation of Reliability per Unit Lifetime.

<table>
<thead>
<tr>
<th>n</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R(n)/L(n))</td>
<td>0.1269</td>
<td>0.1593</td>
<td>0.1759</td>
<td>0.2138</td>
<td>0.2692</td>
<td>0.2797</td>
<td>0.3182</td>
<td>0.3215</td>
<td>0.3241</td>
</tr>
<tr>
<td>n</td>
<td>100</td>
<td>110</td>
<td>120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(R(n)/L(n))</td>
<td>0.3250</td>
<td>0.3449</td>
<td>0.3911</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>130</td>
<td>140</td>
<td>150</td>
<td>160</td>
<td>170</td>
<td>180</td>
<td>190</td>
<td>200</td>
<td>210</td>
</tr>
<tr>
<td>(R(n)/L(n))</td>
<td>0.3896</td>
<td>0.3634</td>
<td>0.3708</td>
<td>0.3773</td>
<td>0.3932</td>
<td>0.3875</td>
<td>0.3763</td>
<td>0.3704</td>
<td>0.3728</td>
</tr>
<tr>
<td>n</td>
<td>220</td>
<td>230</td>
<td>240</td>
<td>250</td>
<td>260</td>
<td>270</td>
<td>280</td>
<td>290</td>
<td>300</td>
</tr>
<tr>
<td>(R(n)/L(n))</td>
<td>0.3763</td>
<td>0.3695</td>
<td>0.3343</td>
<td>0.3348</td>
<td>0.3311</td>
<td>0.3149</td>
<td>0.3086</td>
<td>0.3202</td>
<td>0.3060</td>
</tr>
</tbody>
</table>

Table 7.7 has been divided into three sections for 10 \(\leq n \leq 120\), 130 \(\leq n \leq 210\) and 220 \(\leq n \leq 300\). In the first of these regions the function \(R(n)/L(n)\) may be approximated by

\[
0.12609 + 0.0022226n \tag{8}
\]

or equivalently since this work has determined \(L(n)\) in the previous section

\[
R(n) = 0.12609 \times L(n) + 0.0022226 \times n \times L(n) \tag{9}
\]

for 10 \(\leq n \leq 120\).
This indicates that 12.5% of the *Lifetime* figure represents a lower bound for the *Reliability*, but that the *Reliability* per unit *Lifetime* will increase by about 2% of the *Lifetime* as the number of sensors is increased by 10.

In the next region the rate $R(n)/L(n)$ reaches its highest level and stays constant at about 0.3779, so that the *Reliability* is about 38% of the *Lifetime* figure for $130 \leq n \leq 210$.

In the last region the rate decreases, this can be satisfactorily explained by the fact that the *Lifetime* may be considered to be constant in this region and thus increasing the number of sensors just increases the complexity of communication to the sink node. The rate $R(n)/L(n)$ can be approximated by

$$0.549282 - 0.000832666n$$

for $220 \leq n \leq 300$. So in this region 55% of the *Lifetime* is an upper bound for the *Reliability*, but this decreases by about 0.8% for each extra 10 sensors.
Figure 7.15: (a) R/L Model (Red), R/L Data Points (Blue), 10 – 120 Sensors; (b) R/L Model (Red), R/L Data Points (Blue), 220 – 300 Sensors.
7.6. Radius Model

This section will analyze the effect of Radius and then construct a Radius model. In latter experiments the transmission radius was varied from 1 to 10 metres with \( n = 200 \) or 20. The experiments are then repeated above, but if the nearest neighbour (including the sink node) in the NC protocol to a given sensor sink node is out of range, then not only is the given sensor excluded from the calculations, but also any sensors further out on any path of nearest neighbours leading to the given sensor also have to be excluded. In the case \( n = 200 \) the expected distance between nearest neighbours in the NC protocol is 0.5108, so that in reality varying the radius between 3 and 10 metres can at worst only have a very marginal effect on both the Lifetime and Reliability. In fact, in this case the average number of neighbour sensors in the NC protocol that are greater than two metres apart is 0.003 and so may be considered to be zero, the one-metre case will be considered in more detail below. It has to be borne in mind that this represents a snapshot of the system at the outset. As sensors die two effects have to be considered, the first is that the average distance between nearest neighbours will increase, in turn decreasing connectivity, but it may also be the case that a sensor that was on a path that didn’t reach the sink node (a cul-de-sac) could be rerouted on to a path that does reach the sink node. To confirm these observations the experimental data shows that the Lifetime varies between 236 and 254 with no pattern and the Reliability between 85 and 90% (for radius not equal to one metre) again with no real pattern. These marginal variations may once again be attributed to variations in the topology of the distribution of the sensors, in reality the Lifetime should be considered constant for the radius varying between 1 and 10 metres and likewise for the Reliability, but in this case for the radius varying between 3 and 10 metres.
In the case $n = 20$, the experiments for a varying number of sensors are repeated, but this time with the added complication that the transmission radius can now vary. The average number of nearest neighbours within the transmission radius in the NC protocol is 20 for the radius greater or equal to six metres. Table 7.8 lists the average number of sensors that can transmit along a path to the sink node, using only closest neighbours within the transmission range and the average number of such paths.

For 200 sensors and a transmission radius of one metre, the corresponding figures are 100 sensors and 2.355 trees. (For a transmission radius of two metres the average figure for sensors is 199.945, which may be considered to be 200.)

Table 7.8: Average Number of Trees and Total Sensors on Trees, Limited Radius.

<table>
<thead>
<tr>
<th>Radius</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>0.860</td>
<td>6.796</td>
<td>16.392</td>
<td>19.68</td>
<td>19.98</td>
</tr>
<tr>
<td>Trees</td>
<td>0.556</td>
<td>1.632</td>
<td>2.289</td>
<td>2.547</td>
<td>2.561</td>
</tr>
</tbody>
</table>

So for 20 sensors, the *Lifetime* and *Reliability* may be considered to be unaffected by a transmission radius $\geq 4$ metres. In fact the average number of nearest neighbours that are greater than the transmission radius apart at the outset is 14.8, 6.4, 1.7, 0.3 and 0.04 for transmission radius 1, 2, 3, 4 and 5 metres respectively. Notice in the one-metre case, Table 7.8 indicates that it will often be the case that no sensor will have a nearest neighbour within one metre.

The average distance of a sensor closest (a root sensor) to the sink node is slightly greater than the average distance between closest sensors in the NC protocol, in fact for 20 and 200 sensors this average distance is 1.7008 and 0.5331 metres respectively. However, this implies on a probabilistic basis that root sensors are more likely to
exceed the transmission radius compared to other sensors; if this event occurs all the data from that tree will automatically be lost. In the 200 sensor case there are only on average only about 10 sensors whose nearest neighbour (including the sink node) are greater than one metre apart, but this leads to the loss of data from 100 sensors on average at the outset. This indicates that the Reliability should be about 50% at most for the one-metre case, the slightly higher figure of 56% obtained indicates that some new paths to the sink node will form as sensors die, thus recovering some data. In the two-metre case, the system may be considered to be completely connected at the outset but with the associated increased risk of data collision, but this time when sensors eventually die it will lead to decreased connectivity. This would account for the slightly lower Reliability figure of 85% obtained in this case compared to higher radii.

The quantities in Table 7.8 for 20 sensors with a varying radius are recorded below:

Table 7.9: Expected Distances and Weights, Variable Radius.

<table>
<thead>
<tr>
<th>Radius</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E(d)$</td>
<td>0.3008</td>
<td>1.1100</td>
<td>1.4759</td>
<td>1.5914</td>
<td>1.6175</td>
</tr>
<tr>
<td>$E(d^2)$</td>
<td>0.2247</td>
<td>1.5437</td>
<td>2.6725</td>
<td>3.2117</td>
<td>3.3245</td>
</tr>
<tr>
<td>$w(s)$</td>
<td>0.5704</td>
<td>2.2860</td>
<td>3.2492</td>
<td>3.3229</td>
<td>3.3321</td>
</tr>
</tbody>
</table>

The figures in Table 7.9 for 200 sensors and transmission radius equal to one metre are 0.4789, 0.2818 and 9.372 for $E(d)$, $E(d^2)$ and $w(s)$ respectively.

It is worth noting again that for 20 sensors it will be often the case that no sensors will have a nearest neighbour within one metre and this distorts the figures in Table 7.9. In addition $w(s)E(d^2)$ increases with the radius, so that it is natural that the Lifetime
will decrease with the radius as the distances between neighbour sensors on trees connected to the sink node are bounded above by the transmission radius.

For the Reliability, the main factor will be sensors that cannot find a path to the sink node leading to a total loss of data from these sensors. Sensors that do not have a nearest neighbour within the transmission radius will not attempt to transmit, but may receive transmissions from other sensors for which they are the nearest neighbour. The sensors that remain on paths to the sink node now form a system of reduced complexity. Overall then, the reduced complexity should mean that the sensors on paths to the sink node should have an increased Reliability, but this is far outweighed in the system as a whole for small transmission radii by the total loss of data that cannot find a path to the sink node. So in general it should be expected that the Reliability increases with the radius. Connectivity is thus the dominant factor in the case for small radii, but with a very short number of hops for connected sensors, so that these have a very high probability of transmitting successfully. It can be observed that the number of sensors that have a NC neighbour for radius 1 or 2 metres divided by 20 is approximately equal to the Reliability. Using the figures under Table 7.8 this yields 26% and 68% for radius 1 and 2 metres respectively, which are good estimates of the Reliability. Once 3 metres is obtained, connectivity is no longer a great issue, but the average path length has increased to 3.3, so that data collision becomes more significant.

This chapter has detailed the Nearest Closer protocol and the NC protocol has been successfully modelled mathematically. This thesis (in Section 3.3) has mentioned several location-based routing protocols. Greedy Perimeter Stateless Routing (GPSR detailed in Section 3.3.1) is one of these protocols. In the NC protocol, if the sink node cannot be reached then the data is lost. If it is possible to signal to sensors that a particular route is closed and so always for data to find a route (if one exists) to the sink node (as in GPSR), then the Reliability can be improved. In this case the more energy will be consumed which will lead to a decrease of network Lifetime.
From the Formula 9 of this chapter

\[ R(n) = 0.12609 \times L(n) + 0.002226 \times n \times L(n) \]

for \( 10 \leq n \leq 120 \).

*Reliability* will decrease as the *Lifetime* goes down. So this equation will no longer work for the updated routing protocol. This is one of the drawbacks of this approach.

### 7.7. Summary

In this chapter, deriving from a series of experiments undertaken in the J-Sim simulation tool several conclusions emerge for NC:

(a) The *Reliability* for this application increased as the number of sensors increased to 40, when it reached its highest value. It then essentially decreased as the number of sensors increased from 40 to 300.

(b) The *Lifetime* decreased as the number of sensors increased to 130 and then it fluctuated slightly as the number of sensors increased to 300. The *Lifetime* reached its lowest value when the number of sensors equalled 220, whereas the highest value of the *Lifetime* occurred when there were 10 sensors.

(c) The *Radius* can significantly affect the *Reliability* and *Lifetime*, when there are just 20 sensors. As the *Radius* increased from 1 to 10 metres, the *Lifetime* decreased from 757 to 569. On the other hand, the *Reliability* increased as the transmission radius increased from 1 to 5 metres. Then the *Reliability* fluctuated slightly as the radius increased to 10 metres.
Based on these simulation results, *Reliability, Lifetime* and *Radius* models in the form of equations have been constructed. With these equations, the *Reliability, Lifetime* and *Radius* can be predicted without further simulations.

The next chapter will reflect on the work undertaken within the thesis and will outline some directions for possible future work.
Chapter 8: Critique and Future Work

This chapter reflects upon the main results of this thesis emphasizing the routing protocols studied and equations derived; this will enable possible future work to be considered.

8.1. Critique

8.1.1. Routing Protocols

This thesis has detailed several wireless sensor network routing protocols specifically.

Single-hop Protocol:

This is a simple protocol in which sensors transmit directly in one hop to the sink node. Latency will thus be minimized, but data collision may occur at the sink node and sensors situated a long way from the sink node may expend a large amount of energy in transmitting to the sink node or may not be able to transmit.

LEACH Protocol:

LEACH is representative of the class of hierarchical protocols, in which most sensor nodes transmit sensed data to cluster heads; the cluster heads aggregate and compress the data and in turn forward the data to the sink node. The LEACH protocol uses randomized rotation of cluster heads and also reduces the amount of data that needs to be transmitted to the sink node. Consequently the use of clusters decreases the number of intermediate nodes. Furthermore, using rotating cluster heads and adaptive clusters, the energy requirements of the system are in general distributed among all the sensors.
Each node uses a stochastic algorithm at each round to determine whether it will become a cluster head in that round. This algorithm can be explained as follows:

The decision to act as a cluster head or not is made by node $n$ choosing a random number between 0 and 1. If the number is less than a threshold, $T(n)$, the node becomes a cluster head for the current round. In fact $T(n)$ is set equal to $P / (1 - P * (r \mod (1/P)))$, if node $n$ belongs to the set of sensor nodes that have not been cluster heads in the last $1/P$ rounds and otherwise it is set equal to 0. Here $P$ (with $1/P$ an integer) is the desired percentage of cluster heads and $r$ stands for the current round.

Consequently, if the remaining energy for each node can be measured, this would make a significant contribution to this research area.

**Nearest Closer Protocol:**

The NC protocol relies on the strategy of greedy forwarding, which tries to bring a transmitted packet closer to the sink node in each step or hop using only local information. Thus each sensor forwards the message to its neighbour that is most suitable from a local point of view.

**8.1.2. Evaluation Parameters**

This work has analyzed a number of parameters for the applications in WSNs and defined in particular *Reliability*, *Radius*, *Density* and *Lifetime*.

**Reliability:**

$$Reliability = \frac{\text{the number of packets received by the sink node}}{\text{the number of packets sent to the sink node}}$$
The definition of Reliability makes it easy to analyze communication among the sensors; in particular it makes the estimation of data collision in the wireless sensor network possible.

**Lifetime:**
Network lifetime has become the key characteristic for evaluating sensor networks in an application-specific way. In particular the availability of nodes and the connectivity have been included in discussions on network lifetime.

The definition for network Lifetime, which was taken in the experiments in this thesis, is defined as the time when the last packet is received by the sink node.

**Radius:**
In a wireless sensor network, all the nodes are deployed with a predefined communication radius. This limits the area that the sensors can detect phenomena in, so that this thesis will also evaluate the effect that this limitation on communication radius has on the Lifetime and Reliability by varying the Radius.

**Density:**
The number of sensors deployed in a fixed area was taken as the Density parameter in this thesis. Obviously as the number of sensors increases, so does the average number of sensors per square metre i.e. the density of the sensors.

8.1.3. J-Sim and Simulation Results

**J-Sim:**
This thesis has implemented Single-hop, LEACH and NC routing protocols in J-Sim and obtained several useful simulation results.

J-Sim is an open-source, component-based compositional network simulation
environment. The framework provides an object-oriented definition of target, sensor and sink nodes, sensor and wireless communication channels, and physical media such as mobility models and power models.

The authors of J-Sim have performed detailed performance comparisons in simulating several typical WSN scenarios in J-Sim and NS-2. The simulation results indicate J-Sim and NS-2 incur comparable execution time, but the memory allocated to carry out simulation in J-Sim is at least two orders of magnitude lower than that in NS-2. As a result, testing undertaken within NS-2 often suffered from out-of-memory exceptions and was unable to support large-scale WSN simulations, In contrast J-Sim exhibited good scalability and was demonstrably fit for purpose.

In this thesis, several results have been obtained. The Reliability for the Single-hop application decreases as the number of sensors increases. Although the relationship is essentially linear, the Reliability is above 99% in all cases considered. The reason for above results can be explained by observing that as the density of sensors goes up more sensors will join the data transmission process, and consequently the communication at the sink node will become more complex. Since a large radius is being used all sensor nodes can communicate with the sink node and therefore the only packet losses that can occur are through data collision. The CSMA protocol is implemented to minimize such collisions. The results demonstrate that the Single-hop protocol has a small error percentage associated with it, allowing data collision to occur and this explains why the Reliability is inversely proportional to the Density.

On the other hand the Reliability for LEACH decreased from 22.86% to 1.89% as number of sensors goes up. The reason for this can be explained as follows: As the number of sensors increases, the cluster heads will consume much more energy to communicate with the sensor nodes. Then the remaining energy for each cluster head will decrease rapidly with the increasing Density. In addition, transmitting
packets to the sink node will consume a lot of energy, and then more and more elected cluster heads cannot transmit data to the sink node. Thus it is reasonable to expect the Reliability to decrease with the Density.

The Reliability for LEACH is much lower than that of Single-hop. This is because cluster heads will consume a lot of energy to communicate with the sensors in the clusters. The cluster heads may not have enough energy to transmit packets to the sink node, but they will continue to run and drop packets. Also when data collisions do occur they will be catastrophic with all the data from the cluster heads being lost in one collision, whereas it would only be the data lost from individual sensors in the Single-hop case.

The Reliability for NC protocol increased as the number of sensors increased to 40, when it reached its highest value. It then essentially decreased from 93.64% to 75.89% as the number of sensors increased from 40 to 300, when it reached its lowest value. This decreasing curve can be explained by observing that as the density of nodes increases, more sensors will join the data transmission process and consequently communication among the sensors will become more and more complex. So dropped data due to data collision and latency cannot be ignored. Consequently it is reasonable to expect the Reliability to decrease with the Density. The reason why the Reliability increases as the number of sensors goes up from 10 to 40 has not been discovered in this thesis. This is one of the drawbacks of this thesis. As the data collision for NC is more complex than the Single-hop protocol, the average Reliability is lower than that of Single-hop seems reasonable. The cluster heads data collisions will lead to a much lower Reliability of LEACH than that of NC protocol.

The Lifetime for the Single-hop protocol is always around 15600. Thus the Lifetime is not affected by the density of sensors as the Lifetime for Single-hop is being
predominantly measured by the *Lifetime* of the nearest sensor to the sink node. Increasing the number of sensor nodes will on average decrease the distance of the nearest sensor to the sink node but this decrease will be very marginal in the range of 10 to 60 sensors. Thus it has a very minimal effect on *Lifetime*.

However, in LEACH, the network *Lifetime* is much lower than that for Single-hop. This is because cluster heads will consume energy in communicating with the sensor nodes in the clusters, aggregating data and in transmitting aggregated data to the sink node. The *Lifetime* for LEACH increases from 1541 to 4043 as the number of sensors increases from 4 to 60. This can be explained as follows: As the density of sensors increases, more sensor nodes can be elected as cluster heads and so the energy for each node can be conserved. Thus the *Lifetime* of the network should go up as the density of sensors increases. The relationship between the number of sensors and the *Lifetime* for NC is negative correlation.

The *Lifetime* for NC decreases from 702 to 228 as the number of sensors increases from 10 to 130. The average *Lifetime* is lower than that of LEACH. The reason for these results can be explained as follows: In the NC protocol, sending a packet to the sink node will require each sensor to transmit its data down transmission trees to the nearest sensors to the sink node. These nearest sensors (always two or three) will receive a lot of packets and transmit these packets to the sink node, which will consume a large amount of energy. Thus a NC network will die rapidly from the sink node outwards. If the LEACH network with two or three clusters was run with fixed cluster heads then one would expect that its *Lifetime* would be very short as each cluster head has to collect and transmit all the packets in the network. However, the crucial feature of LEACH is that the cluster heads are rotated amongst the sensors, allowing this energy burden to be spread evenly amongst the sensors and so leading to a much longer *Lifetime*. Then the *Lifetime* for the NC protocol is always around 240 as the density of sensors increases from 140 to 300. The highest value 702 occurs
when there are 10 sensors. The lowest value 223 occurs when there are 220 sensors.

The reason why the Lifetime no longer decreases when the number of sensors is greater than 130 has not been discovered in this thesis. This is one of the drawbacks of this thesis.

8.1.4. Equations

Chapter 5, 6 and 7 have detailed the main results for the Single-hop, LEACH and NC protocols respectively. Based on these simulation results, this work has proposed several important mathematical equations.

For the Single-hop protocol:

\[
\text{Lifetime} = 15600, \quad \text{(Section 5.3.1)}
\]

and

\[
\text{Reliability} = 100.014 - 0.0126857142857145n, \quad \text{(Section 5.4.1)}
\]

where \( n \) denotes the number of sensors.

The actual value obtained for the constant \( \text{Lifetime} \) will of course vary depending on the battery size each sensor in the experiments. This thesis will expect better batteries to increase \( \text{Lifetime} \). If the battery size and power was a variable per sensor then the conclusions of this experiment, of a constant \( \text{Lifetime} \), could no longer be drawn.

For the LEACH protocol:

\[
\text{Lifetime} = 1135.237983n/[1 + 0.24225n], \quad \text{(Section 6.5)}
\]
and

\[ Reliability = \frac{0.396385551}{[2 + 0.326372516(n - 2)]}, \]  
(Section 6.6)

where \( n \) denotes the number of sensors.

For the NC protocol:

Let \( n \) denote the number of sensors, then

\[ \text{Lifetime} = \frac{4918141600.24}{6653n^2 + 6867497n + 125057800}, \]  
(Section 7.4.2)

for \( 10 \leq n \leq 130 \), and

\[ Reliability = 0.12609 \times \text{Lifetime}(n) + 0.002226 \times n \times \text{Lifetime}(n) \]  
(Section 7.5)

for \( 10 \leq n \leq 120 \).

8.1.5. Drawbacks of the approach

The main drawbacks of the approach are as follows:

1. The first disadvantage is that there are no experiments on real sensors within this thesis. The reason why such experiments have not been undertaken is due to the high cost for the large number of sensors that would be required. All the equations obtained in this work are based on simulation results. There probably exist differences between simulated and real results, which can only be rectified by performing some experiments on a test bed of real sensors. The Lifetime of the single-hop protocol is one such example. Simulated experiments indicated that
theLifetimeof the network is not affected by the number of deployed sensors, however the specific constant value found in Section 5.3.1 will differ in reality due to differences in battery capacity, data format, etc. However, the aim of this thesis was to construct mathematical models from the simulations, and it is the author’s intention to carry out experiments on real sensors in the future. Simulation has the advantage of being able to model a large number of sensors, which is hard to implement for real applications.

2. Some real-life applications, such as detection of temperature and moisture in a vineyard, would have sensors placed rather than randomly positioned. The simulations in this thesis do not cover this scenario, although placed sensors should generally perform better than randomly distributed one from the viewpoint of the user.

3. Although the present work is useful for comparing a representative example of each protocol type, further work to compare different examples within the same protocol family would be useful in future.

4. The existing equations may not support updated representative routing protocols. So further work need to be done in order to construct the new models for the updated representative routing protocols.

5. The setting of the simulation area is fixed and the sink node is located in the middle of this area. The reason for this set-up is that it has military (and other) applications, such as a warship at sea (the sink node) deploying sensors to detect another vessel. Another obvious choice for the position of the sink node is in one of the corners, but placing the sink node in the centre utilizes the symmetry of the square deployment area. So the equations obtained in thesis have limitations as well.

6. There may be limitations to which the results are scalable, if the network area was
increased to a 100 by 100 metre square, the results for a small number of sensors in such a large area might demonstrate some new behaviour.

8.2. Future Work

A flat protocol (Single-hop), a location-based protocol (Nearest Closer), and a hierarchical protocol (LEACH) have been detailed and these protocols have been implemented into the J-Sim platform. Future work will focus on the construction of evaluation models for more routing protocols and the implementation of the mathematical model.

8.2.1. Power Management

In order to implement power management, future work will focus on integrating J-Sim’s energy model into SunSpot sensors. Agent Factory Micro Edition (AFME) [119–121] is an agent-based programming implemented in Java compatible with being run on the Sunspot sensor.

AFME was developed to enable the creation of intentional agents for mobile devices. The previous version of AFME was Agent Factory, which was implemented in Java. As the code works on mobile sensors or other mobile terminals, AFME is implemented in J2ME. (J2ME is one of the popular programming languages for mobile devices.)

Within AFME some crucial functions are based on several perceptors and actuators. The perceptors generate beliefs about the system state and the environment; according to this information and its internal declarative rule set, the agent will decide on the plans or actions that need to be performed.
SunSpot sensors are Java-based and a number of routing protocols can be set into these sensors as well. In trying to use J-Sim in conjunction with SunSpot sensors several barriers will need to be overcome and are detailed below:

(a) The main idea for J-Sim’s energy model is to set up a battery for each sensor. As sensors consume energy via communication, they will let the battery know the energy consumption. Thus the battery can calculate the remaining energy for each sensor. Consequently setting up a battery calculation mechanism for SunSpot sensors is necessary. In most instances the users are not concerned about the protocol adopted, but are merely interested in the results obtained. Ideally the system would be fully automated with the agents able to implement a change in protocol without any outside help.

(b) The second issue is to estimate the communication cost. A sensor may be in transmitting state, receiving state, or sleeping state, so communication in different states will consume different levels of energy. In order to implement power management, estimating this energy consumption becomes a significant problem.

(c) The third issue is to integrate J-Sim’s energy model into SunSpot sensors. As J-Sim is a Java-based simulation tool, the energy model for J-Sim can be converted into a J2ME program and integrated to SunSpot sensors. However, there exist several practical problems in the implementation of such work. One problem is that if the energy calculation mechanism is incorporated within Sunspot sensors, the calculation itself will also consume energy. It is hard to estimate this energy consumption and it cannot be ignored as the calculation is very frequent. If one calculation costs more energy than one communication, that would be very poor.

If these problems are solved, then the users would know the energy status of each sensor and change the routing protocol or parameters as necessary.
8.2.2. The Construction of Evaluation Models for More Routing Protocols

To integrate additional complex routing protocols into J-Sim is both possible and also very significant for potential future work.

What are the routing protocols that can be integrated in J-Sim? This thesis has implemented a flat protocol (Single-hop), a hierarchical protocol (LEACH) and a location-based protocol (Nearest Closer) within wireless sensor networks. So the future work will focus on the extension of these three primary types of routing protocols. Chapter 3 has detailed several routing protocols in each type. It would be desirable to implement the flat routing protocols (Sensor Protocols for Information via Negotiation, Rumor Routing), hierarchical routing protocol (Threshold-sensitive Energy Efficient protocol), and location-based routing protocols (Greedy Perimeter Stateless Routing protocol, Nearest with Forwarding Protocol).

The Threshold-sensitive Energy Efficient protocol is a more complex protocol than LEACH, although similar to it. The Threshold-sensitive Energy Efficient protocol has not been implemented in J-Sim to date, but given its similarity to LEACH this should be achievable in the future.

An attempt has been made within J-Sim to integrate the Greedy Perimeter Stateless Routing protocol, unfortunately in reality this does not work. The programming of this complex routing within J-Sim contains a fault (or faults), which could be rectified.

On the other hand, the implementation of the Nearest with Forwarding Protocol into J-Sim should be relatively straightforward given that the Nearest with Forwarding Protocol is similar to the Nearest Closer protocol.

Based upon these routing protocols, more simulation results could be obtained for
different routing protocols, and thus some new evaluation models could be constructed. Then the researchers could compare the models that were obtained in this thesis to the new models and consequently the evaluation models will make more sense.

If some more evaluation models could be constructed, a user interface will be required. The user interface will allow an end-user to input the desired values of parameters from the network, such as requiring that the network has a long life but low accuracy, and the model will then suggest the routing protocol, number of sensors and so on to be used in the network.

8.2.3. Mathematical Model Implementation

Once again it will be supposed that the SunSpot sensors have been set with a number of routing protocols. Consequently, agents in the sink node, equipped with knowledge of the equations obtained in this thesis, would be able to recommend the most suitable routing protocol to the user. Thereafter the user can set this selection information into a given sensor prior to deployment. This sensor would subsequently broadcast this information to all the other sensors in the sensor network; this may consume a lot of energy for this one sensor, but it can be sacrificed. After the deployment of these nodes the energy consumption of each sensor can be transmitted to the sink node and then obtained by the users. Consequently the users can control and analyze the energy consumption of this sensor network.

As previously mentioned there are problems integrating routing protocols into SunSpot sensors. It is also the case that AFME presently can only manage simple reactions, in the sense that it only offers a limited number of commands. On the other hand AFME is the only Java-based programming language that works on SunSpot sensors. In addition, at present, insufficient mathematical models exist which could inform protocol choice.
8.2.4. Other related work

In the NC protocol, the sink node is in the middle. Tree roots are close to the sink node and they will die quickly as they do receive and transmit data frequently. The battery capacity is the same for all sensors in the experiments. If the tree shape can be predicted, then it would make more sense to have more power capacity at the tree roots in a real deployment.

8.3. Summary

This chapter has reviewed this thesis and provided some details about future work. The future work will focus on power management, the construction of evaluation models for more routing protocols, and the implementation of mathematical models. This future work will be very useful as it will provide users with details of energy consumption for real sensors. As a consequence, users will then be able to select the best routing protocol from various possibilities for the sensor network without any simulation work. The next chapter will conclude this thesis.
Chapter 9: Conclusions

Flat protocols, hierarchical protocols and location-based protocols constitute the three primary types of routing protocols within wireless sensor networks. Single-hop, LEACH and Nearest Closer are representative and basic routing protocols for each of these types respectively. Consequently within this thesis, Single-hop, LEACH and Nearest Closer protocols have been integrated into the J-Sim simulation tool.

Energy consumption (measured within this thesis by the Lifetime), Density, Radius, and Reliability are key parameters within wireless sensor networks.

A series of simulation results have been presented within this thesis. Based on the results, this thesis has developed some mathematical models that inform the design of more effective and efficient wireless sensor networks without prior use of simulations. The most important formula that has been obtained is the following one in LEACH:

\[ \frac{\text{Reliability}}{\text{Lifetime}} \times \text{Number of sensors} = \text{Constant} \]  

(Section 6.4.1)

This equation says that the rate of successful packet reception per unit time is independent of the number of sensors. The constant in the equation will depend on the parameters in the simulation and it is a reasonable model if the number of cluster heads is small. The number of sensors in the cluster is one way of measuring packets received, but again ignores data lost from sensors before it reaches the cluster head. Such lost data will increase with the number of sensors, so that the constant on the right-hand side of the equation may have to be found for a small, medium or large number of sensors.

151
This thesis has encoded and integrated java implementations of the Single-hop, Nearest Closer and LEACH protocols into the simulation tool J-Sim and provided a mathematical model for each protocol. These mathematical models capture relationships between certain key parameters, which in most cases are linear or piecewise linear. The models also illustrate that in some cases a parameter, such as Radius, is not generally critical to the values obtained for other parameters, such as the Lifetime and Reliability, which enables users to disregard such redundant parameters. After more protocols and their associated models have been added, more parameters can be predicted directly. The evaluation models within this thesis allow the deployment of sensor nodes to be accomplished without simulations to satisfy the demands of the customer, resulting in a significant saving of time and expenditure.
References


