ENERGY EFFICIENT HYBRID ROUTING PROTOCOL FOR WIRELESS SENSOR NETWORKS

JG PAGE

2007

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

ENERGY EFFICIENT HYBRID ROUTING PROTOCOL FOR WIRELESS SENSOR NETWORKS

By

Jonathan Grant Page

(21255637)

Studyleader: Professor G.P. Hancke

Submitted in partial fulfillment of the requirements for the degree

Master of Engineering (Computer)

in the

Department of Electrical, Electronic & Computer Engineering

in the

School of Engineering

in the

Faculty of Engineering, Built Environment & Information Technology

UNIVERSITY OF PRETORIA

October 2007

SUMMARY

ENERGY EFFICIENT HYBRID ROUTING PROTOCOL FOR WIRELESS SENSOR NETWORKS

by

Jonathan Grant Page

Studyleader: Professor G.P. Hancke

Department of Electrical, Electronic & Computer Engineering

Master of Engineering (Computer)

A wireless sensor network is designed to monitor events and report this information to a central location, or sink node. The information is required to efficiently travel through the network. It is the job of the routing protocol to officiate this process. With transmissions consuming the majority of the energy available to a sensor node, it becomes important to limit their usage while still maintaining reliable communication with the sink node.

The aim of the research covered in this dissertation was to adapt the flat and hierarchical architectures to create a new hybrid that draws on current protocol theories. The designed and developed protocol, Hybrid Energy Efficient Routing (HEER) protocol, builds upon the initial groundwork laid out by the previously developed Simple Energy Efficient Routing (SEER) protocol designed by C.J. Leuschner. Another aspect of the work was to focus on the current lack of credibilty that is present in the WSN research community. The validity of SEER was examined and tested and this led to the main focus of this research, ensuring that HEER proves to be valid.

The HEER protocol for wireless sensor networks is designed such that it is computationally simple, limits the number of transmissions, employs a cross-layer approach, is reliable, is energy-aware, has limited support for mobile nodes, is energy efficient, and most importantly is credible.

Sensor nodes are extremely limited when it comes to their available energy resources. To maximise the node and network lifetimes requires the designed algorithm to be energy aware and as efficient as possible. A cross-layer design approach is followed which allows for the different layers of the OSI model to interact.

The HEER protocol limits the number of transmissions that are used for network operation. This is achieved by using a minimal amount of messages for network setup and by selecting the optimal route. Route selection is calculated using hop count, current energy available, energy available on the receiving node, and lastly the energy required to reach the destination node.

HEER combines and expands upon the method used by SEER for route selection. Network lifetime for networks of large sizes is increased, mainly due to more efficient routing of messages. The protocol was kept computationally simple and energy efficient, thus maintaining network survivability for as long as possible.

Keywords:
Flat Routing, Hierarchical Routing, Wireless Sensor Networks, Energy Consumption, Energy Efficiency, Node lifetime, Clustering.

OPSOMMING

ENERGY EFFICIENT HYBRID ROUTING PROTOCOL FOR WIRELESS SENSOR NETWORKS

deur

Jonathan Grant Page

Studieleier: Professor G.P. Hancke

Departement Elektriese-,Elektroniese- & Rekenaar Ingenieurswese

Meester in Ingenieurswese (Rekenaar)

’n Draadlose sensornetwerk (DSN) word gebruik om parameters en gebeurtenisse te monitor en hierdie inligting aan ’n sentrale punt, of putnode, te rapporteer. Dit is noodsaaklik dat hierdie inligting effektief deur die netwerk beweeg en hierdie aspek word deur die roeteringprotokol hanteer. Aangesien transmissies die meeste van die sensor-energie verbruik, is dit belangrik om dit te beperk maar steeds betroubare kommunikasie met die putnode te handhaaf.

Die doel van die navorsing in hierdie verhandeling beskryf was om die plat en hiërargiese argitekture aan te pas om ’n nuwe hibried gebaseer op huidige protokolteorieë te skep. Die protokol wat ontwikkel is, die Hibriede Energie-Effektiewe Roeterings-protokol (HEER), bou voort op die grondslag wat gelê is deur die Simplistiese Energie-Effektiewe Roeterings-protokol. Voortvloeiend uit hierdie werk is gekonsentreer op die geldigheidsaspek van die bevindinge wat in die DNS-navorsingsgemeenskap gepubliseer is en word. As deel hiervan is die geldigheid van SEER krities ondersoek en getoets. ’n Belangrike doelwit van hierdie navorsing was om te verseker dat die geldigheid van HEER bevestig word.

Die HEER-protokol vir draadlose sensornetwerke is ontwerp sodat dit berekeningsgewys eenvoudiger is, die getal transmissies beperk, ’n kruislaagbenadering gebruik, betroubaar

is, energiebewus is, beperkte ondersteuning bied aan mobiele nodes, energie-effektief is en bowenal geldig is.

Sensornodes het ’n beperkte energiebron. Om die lewensduur van die node en netwerk te maksimeer, is dit nodig dat die algoritme wat ontwerp word energiebewus en so doeltreffend moontlik is. Vir hierdie ontwerp is ’n kruislaagbenadering, wat interaksie tussen die verskillende lae van die OSI-model toelaat, gevolg.

Die HEER-protokol beperk die aantal transmissies wat vir die funksionering van die netwerk gebruik word. Dit word bereik deur die minste moontlike aantal boodskappe vir die opstelling van die netwerk te gebruik en die optimale roete te kies. Die roetekeuse word bepaal deur die gebruik van die hoptelling, die beskikbare energie op die nodes, die beskikbare energie op die ontvangsnode en laastens die energie wat beskikbaar is om die bestemmingsnode te bereik.

HEER kombineer en brei uit op die metode wat deur SEER vir die roetekeuse gebruik word. Die netwerkleeftyd van groot netwerke word verleng, hoofsaaklik deur meer doeltreffende roetering van boodskappe. Die protokol is berekeningsgewys eenvoudig en energie-effektief gehou om daardeur so lang moontlik netwerkleeftyd te verseker. .

Sleutelwoorde:
Plat Roetering, Hiërargiese Roetering, Draadlose Sensornetwerke, Energieverbruik, Energie-effektiwiteit, Nodeleeftyd, Groepering.

To Sarah and my Parents

who supported me through my years of study and during my work on this dissertation

ACKNOWLEDGEMENT

Foremost, a special word of thanks should go to my study leader and promoter, Professor Gerhard Hancke, for his support and guidance during my studies.

Without the financial support of Telkom SA Ltd, and their Centre of Excellence programme, I would not have had the opportunity to complete this research and my studies, for this I thank them.

I would also like to mention the following people for their support, and allowing me to bounce many of my ideas off of them; J. Van Wyk, C. Tönsing and J. Hinds.

A special thank you goes out to M. Ferriera and M. Bekker. These two wonderful ladies kept me on the straight and narrow and helped with many an administrative task.

Lastly I would like to say thank you to my wife, Sarah, for looking after me while I tirelessly worked on this research. Without your love and support, I would not have been able to complete this work.

CONTENTS

LIST OF ABBREVIATIONS

3G	3rd Generation
ACK	Acknowledgement Message
ADV	Advertisment Message
AODV	Ad-Hoc On-demand Distance Vector Routing
AP	Access Point
BSS	Basic Service Set
CCK	Complementary Code Keying
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
DARPA	Defense Advanced Research Projects Agency
DSSS	Direct Sequence Spread Spectrum
FCC	Federal Communications Commission
FFD	Full Function Device
FHSS	Frequency Hopping Spread Spectrum
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HEER	Hybrid Energy Efficient Routing
HID	Human Interface Device
HL2	HiperLAN/2
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronic Engineers
IP	Internet Protocol
IR	Infra-Red
ISM	Industrial, Science and Medical Band
ITU-R	International Telecommunication Union - Radiocommunications Sector
JOIN	Joining Message
Kb	Kilobyte
LAN	Local Area Network
LBNL	Lawrence Berkley National Laboratory

ABBREVIATIONS

LMDS	Local Multipoint Distribution System
LEACH	Low-Energy Adaptive Clustering Hierarchy
LOS	Line of Sight
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MF	Mobility Framework
MTE	Minimum Energy Tranmission Protocol
NAM	NS-2 Network Animator
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open System Interconnect
PAN	Personal Area Network
PARC	XEROX Palo Alto Research Center
PC	Personal Computer
PDA	Personal Digital Assistant
PHY	Physical
PLE	Path Loss Exponent
POW	Power Message
PRNG	Psuedo Random Number Generator
QoS	Quality of Service
RAM	Random Access Memory
REQ	Request Message
RFD	Reduced Function Device
RSSI	Received Signal Strength Indicator
SEER	Simple Energy Efficient Routing
SPIN	Sensor Protocols for Information via Negotiation
SNR	Signal to Noise Ratio
TCP/IP	Transport Communication Protocol / Internet Protocol
TDMA	Time Division Multiple Access
TTL	Time To Live
UWB	Ultra-Wide Band
VINT	Virtual InterNetwork Testbed
WAN	Wide Area Network
WANET	Wireless Ad-hoc Network

ABBREVIATIONS

WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSAN	Wireless Sensor / Actor Network
WSN	Wireless Sensor Network
xDSL	Digital Subscriber Line

CHAPTER ONE
RESEARCH OVERVIEW

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."

MARK WEISER

1.1 INTRODUCTION

Globalization is evident around us and is increasing on a daily basis. Interesting and new ways are being discovered to locate and communicate the information that is required for every day use. An *Internet of Things* [1] is being established where items such as toothbrushes communicate with a shopping list storage device indicating the need for replacement. These items or products are becoming "aware" and are reporting their information. Another idea that has been around for a number of years is that of *Smartdust* [2]. The idea is to have small devices or nodes that can be found anywhere in the environment. These devices could be dropped from an aeroplane or distributed like plant seeds. The idea could even be further expanded to such an extent where every square metre of the Earth has a number of small nodes. An example of such a node can be seen in *Fig.* 1.1. This figure shows the possible complexity that a wireless sensor node can have and the possible size of the packaging.

Wireless sensors and their applications themselves have evolved over time. The possible applications for these networks are limitless. The early adopters of this technology include the defence and emergency services. From underwater sensors used to track Russian submarines to tracking the moisture and sunlight levels of a vineyard for the perfect grape. The network could provide life saving critical information at critical times. For example, the network could provide information to fire fighters as to the location of a fire and direct them

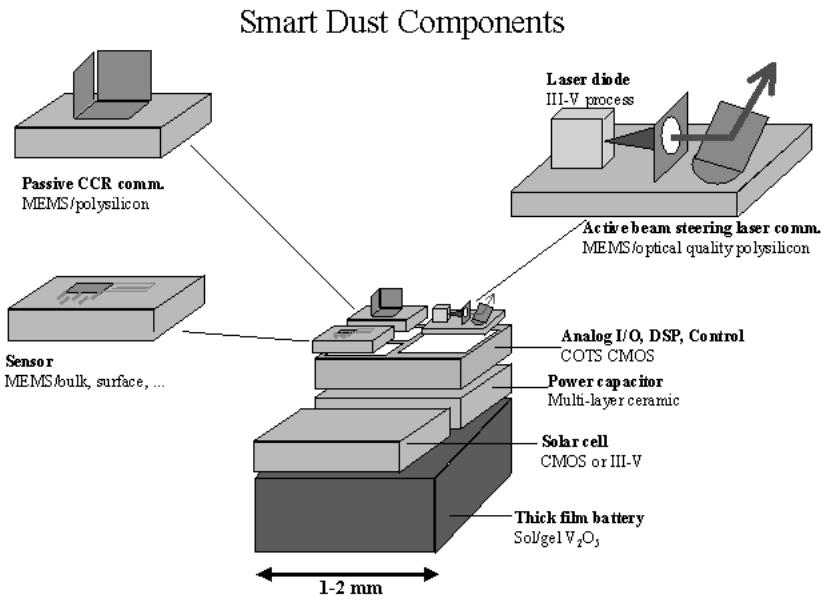

FIGURE 1.1: Proposed example of a Smartdust node. Taken from [2].

to the safest route to it. To create a scenario such as this, these small sensors will need to be able to reliably communicate with each other. Wireless Sensor Networks (WSN) and their protocols were designed specifically for these types of tasks. To achieve these fantastic ideas, these nodes would need to be small, inexpensive, relatively disposable and energy efficient. Each of these unique requirements add constraints on the operation of a node. A limited power supply that may not be able to be replaced easily further complicates their operation. With the need for the networks to communicate with a central location, or base-station, the nodes will need to route their information over multiple nodes.

Networks comprising of thousands of nodes would thus add up to thousands of messages. The overall lifetime of a network of this nature would not be very long. To extend the lifetime, various protocols at all the OSI levels have been designed to achieve better lifetime performance. The network may even be dependant on specific critical nodes. Should these nodes fail, the network would become segmented and some parts may even become unreachable. Thus individual node and overall network lifetime needs to be prolonged to its fullest extent. All of these necessities need to be designed into any protocol or algorithm for the WSN environment. Routing protocols for WSNs are usually designed on one of two main architectures, namely flat or hierarchical. With this in mind, this research focuses on the design and development of a hybrid layer 3 or routing protocol for a wireless sensor

network that uses advantages from both the main routing architectures.

1.2 SCOPE

The scope for this research is the design, development and implementation of an energy efficient hybrid routing protocol for WSNs. The designed routing protocol combines features of both flat and hierarchical routing architectures and implements some cross-layer ideas. The goal is to achieve a generic routing protocol that is scalable, energy efficient and can meet the requirements of any arbitary application layer.

There are many routing protocols for WSNs. Most of these are either in the flat or hierarchical camps. The development of both flat and hierarchical protocols is popular and many exist. Research into hybrid protocols however has been limited, thus the designed protocol adds hierachical features to a flat protocol, in this case the previously developed Simple Energy Efficient Routing protocol (SEER).

1.3 PROBLEM STATEMENT AND RESEARCH CONTEXT

Wireless sensor networks are fast becoming evident in all works of life. This prevalence requires that the network be efficient in handling data as well as being able to act on the information by either directing it to the correct location or acting appropriately. The types of routing protocols currently used each have their benefits as well as their problems. One of the problems in WSNs is that they are still relatively new to the community and common standard protocols that can satisfy all the multitude of tasks does not exist at present. The TCP/IP suite of protocols would be an example of a standard that almost all computers use, especially when communicating on the Internet. A similar WSN protocol that can be used widely and efficiently would allow greater flexibility in the wireless sensor environment.

The problem addressed in this proposal is to design and develop a routing protocol that has as many of the advantages and as few of the disadvantages of the current protocols and to implement these changes into a single protocol. Most protocols written for a WSN only allow for fixed sensors but a WSN would most likely contain a limited number, if not consist only of mobile nodes. If a sensor is mobile, attached to a person for instance, the static

network would be constantly trying to reorganise the network routing structure. Thus further investigation should be conducted to allow for a more flexible routing protocol that can adapt mobility into its environment as effectively as possible.

Wireless sensor nodes themselves are often only provided with a limited power supply. Making efficient use of this available power supply requires all the layers of the device to work in harmony to maximize the lifetime of the node, which in turn increases the lifetime of the network as a whole. This concept is termed as using a cross-layer design.

1.4 RESEARCH OBJECTIVES

The scope and objective of this proposed research is the design and development a WSN routing protocol that can be implemented on existing WSN infrastructures and which exhibits the following criteria:

- The protocol should be scalable and function effectively for networks of any size.
- The protocol has to minimize the computational complexity for the nodes, thus extending the lifetime of the network.
- The protocol must be as simple as possible and as independent from the hardware capabilities of the nodes as possible.
- The protocol must limit the number of required transmissions, thus extending the lifetime of the network.
- The protocol must allow for mobile nodes to traverse the network.

The above objectives of this routing protocol can be summarized as follows; a protocol that allows for scalability, energy efficiency, mobility and simplicity. To achieve these objectives, a number of sub-objectives needed to be completed:

- Examine the current state of WSNs and the possible future that they may provide. Any applications that a WSN may be deployed in will be identified as well as their generic requirements. Once these requirements have been determined they should be investigated as to what effect they will have on the protocol and the WSN.

- Investigate the multitude of routing protocols that are available.
- Identify advantages and disadvantages of the various routing architectures being employed.
- Examine currently avaiable simulators for WSNs.
- Investigate the credibility of research to date.
- Propose and test a new protocol HEER.

1.5 RESEARCH APPROACH

This section identifies the important questions that require solutions, as well as the research approach that was taken to find them.

1.5.1 Research Questions

The following problem solving questions were defined to better understand the problem at hand.

1. What is the current state of routing protocols in WSNs?
2. What architecture would prove the most efficient, and would a hybrid design provide any benefits?
3. Can these benefits be applied to any existing protocol, in this case SEER?
4. Can the number of transmissions be reduced by introducing a hybird approach?
5. Is localization a requirement of a WSN routing protocol?
6. Can the current research results produced via simulations for current routing protocols be trusted?

1.5.2 Research Instruments

1. Literature Study
   (a) The scope of the work was identified.

(b) Relevant work from previous research was identified.

(c) Current wireless technologies were also included

(d) A literature study was undertaken on current flat and hierarchical routing protocols.

(e) A literature study was undertaken to investigate the credibility of WSN simulators and the results produced by them.

(f) Current wireless technologies were also investigated briefly.

(g) A search was done to identify open-source simulators that could be used.

2. Problem Solving Analysis

(a) An analysis of each of the applicable routing protocols structure and operation was undertaken and certain possible benefits were identified.

(b) An analysis of WSN simulators was investigated to determine the most applicable.

3. Design

(a) A routing protocol was designed, Hybrid Energy Efficient Routing, or HEER.

(b) The routing protocol follows a flat architecture and implements some hierarchical features and benefits.

(c) Node mobility was designed into the protocol, then removed as the mobility features extremely degraded performance.

4. Implementation

(a) The HEER protocol was programmed in an appropriate programming language.

(b) HEER was implemented in a suitable simulation environment.

(c) Results for random and uniform networks were obtained.

(d) The protocol was also analysed numerically.

5. Analysis and Assessment

(a) HEER was compared to three other routing protocols:

i. Flooding,

ii. Simple Energy Efficient Routing Protocol (SEER), and

iii. Low-Energy Adaptive Clustering Hierarchy (LEACH).

(b) These routing protocols were also simulated and numerically analysed.

(c) Results are presented and discussed.

(d) Final conclusions are given.

1.6 ORGANISATION OF THE DISSERTATION

This dissertation is organized in the following way:

Chapter 1 Research Overview: Describes the problem and the research approach to finding a viable solution

Chapter 2 Literature Study: Provides background information on relevant wireless sensor network technologies.

Chapter 3 Routing Protocols: The different types of routing protocols that had an impact on this dissertation, their advantages and disadvantages, and the type of applicable simulators that can be found in the WSN field are briefly covered in this chapter.

Chapter 4 Credibility: The current state of WSN reseach, the problems regarding their validity and implementation concerns are covered. A brief look into the credibility of SEER is also conducted.

Chapter 5 Protocol Design: Outlines possible solutions and describes how they can be implemented into HEER, the routing protocol designed for this research.

Chapter 6 Protocol Evaluation: Shows the experimental procedures used to verify and quantify the performance of the proposed implementations. The points made in Chapter 4 are taken into account into designing the simulation environment.

Chapter 7 Results and Discussion: The results achieved during testing and evaluation are documented and explained in this chapter.

Chapter 8 Conclusion: This chapter places the proposed design and the findings from the results in context. Some ideas for further research and expansion of HEER are also mentioned.

CHAPTER TWO
LITERATURE STUDY

2.1 CHAPTER OVERVIEW

With WSNs becoming popular, the reseach into this field has expanded to include all relevant topics imaginable. To understand the research and WSNs, a small overview of the general operations and technologies needs to be discussed.

2.2 OVERVIEW OF WIRELESS SENSOR NETWORKS

2.2.1 Wireless Sensor Nodes

Wireless sensor networks are made up of small nodes or motes[1] usually consisting of a small wireless transceiver, a microcontroller, batteries and some type of sensor and/or actuator. Initially nodes were only configured with a sensor, but the advent of small efficient actuators has resulted in them also being added to the nodes' configuration. A network with "actors" as well as sensor nodes is refered to as a wireless sensor / actor network (WSAN). An example of a sensor node's architecture is shown in *Fig.* 2.1.

The software found on these nodes is usually simplistic in nature. In research and practice, the software for nodes are simplified to only include certain of the OSI layers used in traditional network approaches. The upper layers, higher than the network layer, are amalgamated into a single application layer. This layer is then responsible for all the combined layers. This can be seen in *Fig.* 2.2.

[1] Sensor nodes are occasionally referred to as motes in some research literature.

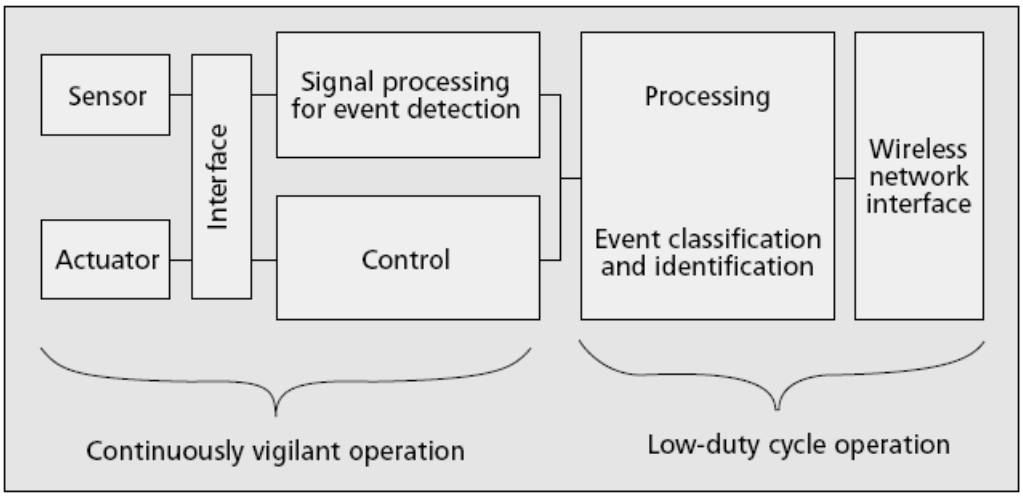

FIGURE 2.1: Typical architecture of a wireless sensor node. Taken from [3] (Fig. 1.).

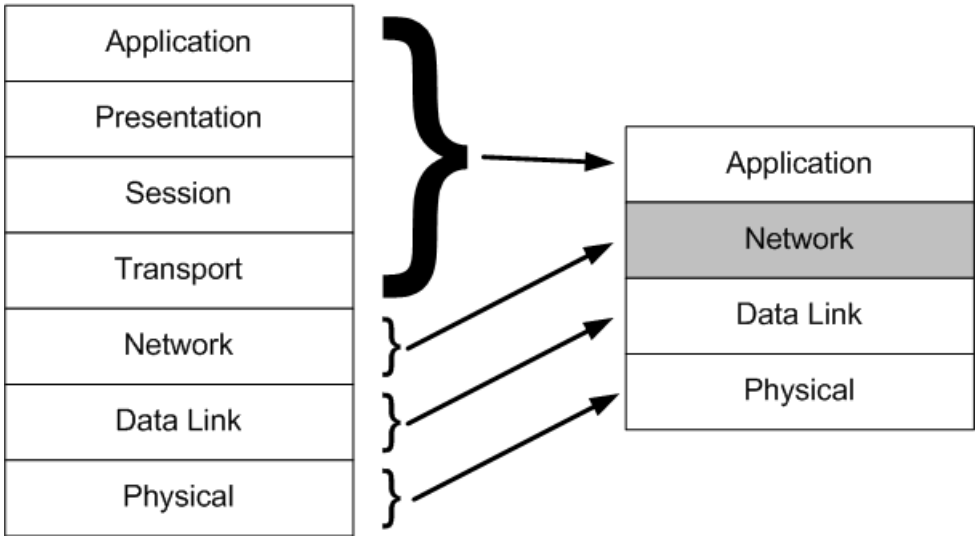

FIGURE 2.2: Protocol layers for WSN as adapted from the OSI model.

2.2.2 WSN vs Ad-Hoc Networks

WSNs have their own unique requirements and differ from tradiontal ad-hoc networks. A sensor node in general serves a unique need in an application where ad-hoc nodes are usually found to be more generic. Some examples of an ad-hoc device could include a PDA, laptop PC or a cellular phone. As mentioned in [4, 5], the key distinguishable differences are:

- *Network Size:* Wireless sensor network sizes can run into the thousands, perhaps in the future even into the millions, e.g. (*Smartdust*). An ad-hoc network on the other hand, usually consists of a smaller number of nodes. One aspect that could also be included in this area is the problem of localization. If a sensor network is deployed over a large geographical area there may exist a need to know the location of that node, else the sensed information may be useless. The two predominate means of determining location is via either GPS or triangulation. Methods using triangulation are better suited for WSNs, as they don't require another power consuming module attached to the sensor node. The transceivers in most provide information that will allow triangulation to a certain accuracy. GPS modules in most cases provide for more accurate positioning but consume large amounts of power.

- *Node Density:* Some applications in the WSN area only require a small number of nodes covering a specific area, but the idea is that there would be some form of redundancy where the ratio of nodes per square metre is larger than that of a traditional ad-hoc network. A node in an ad-hoc network is usually much larger than a sensor node of a WSN and thus the power source and transmission equipment can be greater. This fact alone allows for more sensor nodes to be placed in an area as opposed to ad-hoc nodes. Wireless sensor nodes may be placed in an overlapping pattern so as to maintain network integrity should a neighbour node die.

- *Node Proneness to Failure:* The energy that is available to an ad-hoc node is usually replenishable via recharging or swapping out the batteries. This is not always the case or even an available option with a sensor node. Unless some form of power scavenging is performed, a sensor node is left with the energy reserves that it was given at the start. Power is not the only reason for a possible failure, as mentioned previously, the sensor nodes are smaller in size. This makes them more prone to hazardous environmental conditions when placed randomly and without fore thought,

i.e dropping from an aircraft. Their size would even allow them to be stepped upon after deployment. If a node is placed in a location on purpose and out of harms reach, the only failure should be due to power or equipment.

- *Frequency of Topology Changes:* Due to a node failure, movement or some environmental factors, a sensor network can have many changes to its topology. Routes appearing or disappearing places the most pressure on the network layer and the routing algorithms. A topology change will usually result in some transmissions, and as mentioned these transmissions are the main use of the available energy given to a node. Ad-hoc nodes usually join a network and for the most part can communicate with the gateway or final destination, e.g. sink node, directly. The failure rate is lower and ad-hoc nodes are not usually affected by environmental hazards to the extent that sensor nodes may be.

- *Communication Paradigm Employed:* Communication between wireless nodes are broadcast and are received by all nodes within reception range. Depending on the message address, or lack there of, a message may be discarded or dealt with. Broadcast messages may be used for initial network setup (LEACH and SEER), or for all data transmissions (Flooding). The designed algorithms, MAC and the network layer, need to decide whether to take this functionality into account. Ad-Hoc networks usually use point-to-point communication as they are in direct contact with their sink node. An ad-hoc node also has a more intelligent and powerful routing protocol. A wireless node does not always know the path to a destination, and makes use of broadcasts to send and receive. Another distinct differentiator is that a WSN usually has one destination for communication, the sink, whereas an ad-hoc network may have many destinations for a single transmission.

- *Resource Limitations of Nodes:* The two main limitations for a sensor node include the available bandwidth and the afore mentioned energy reserves. The available bandwidth of an ad-hoc node is usually greater (11 Mbps) than that of a sensor node (250 kbps). All the components are smaller and/or more limited than their counterparts on ad-hoc nodes. Some of these components include; memory, processor, transceiver and batteries. A sensor node could operate using a 4 MHz microcontroller and an ad-hoc node could make use of the latest Intel CPU.

- *Node Identification:* With the possiblity of thousands of nodes, an addressing scheme becomes very important. For possible and effective communication of data, each node would require a unique address. With the numbers so high, it is not possible to have a global addressing scheme and it would be difficult to enforce. Ad-hoc nodes in contrast generally make use of IP addresses which have well defined criteria for their selection and implementation.

Each of these previously mentioned differences add to the woes of designing and implementing the protocols and algorithms for a WSN environment however they must be considered.

2.2.3 Routing Challenges

Having discussed the differences between WSNs and ad-hoc networks, the influencing factors or challenges [4–6] specific to the design of wireless sensor nodes and their algorithms are listed next.

- *Reliability:* Reliability in a WSN refers to the possibility of the network performance being degraded due to node failure. Sensor nodes are more prone to failure, mainly due to the available power. This failure should ideally not affect the functioning of the network. Reliability is calculated as follows [5, 7] using a Poisson distribution. Probability of a node not failing within the interval $(0, t)$ is:

$$R_k(t) = e^{-\lambda_k t} \tag{2.1}$$

  where λ_k is the failure rate of node k and t is the time period.

- *Quality of Service:* Some messages in a WSN need to arrive at their destinations within a prescribed period, else they are no longer valid. Bounded latency is a requirement that energy aware routing protocols should take into account. In many applications, the conservation of energy is more important than the quality of the data sent. At some point a decision would need to be made to reduce the quality of results in order to reduce energy consumption. The reliability of a node (discussed above) is also related to the quality of service.

- *Scalability:* An important aspect to consider when designing protocols, is take the possibility of having thousands of nodes in a network. This together with the likelihood that more nodes could be added at any stage, protocols and routing schemes should be scalable. The node density can be calculated as [8]:

$$\mu(R) = \frac{(N * \pi * R^2)}{A} \tag{2.2}$$

  where N is the number of nodes in region A and the transmission range of the transceiver is R. The result being the number of nodes in transmission range of a specific node.

- *Production Costs:* The nodes used today are still quite expensive (Xbow MicaZ[2] = $\$125$) when compared to the possible requirement for millions of nodes. Nodes will need to cost less the $\$1$ if this lofty goal is to be achieved. [4].

- *Hardware Constraints:* Some design protocols place requirements on the node hardware, and vice versa, some node hardware places constraints on the protocols layers. A high power transmitter for communicating with the sink node would be an example of this [9]. These constraints need to be kept in mind when designing for specific architectures.

- *Network Dynamics:* In much of the research the sensor nodes are assumed to be stationary, but this is not always the case. Many applications call for mobile nodes that operate at various speeds. Routing messages around a network of mobile nodes increases the difficulty associated with topology stability in addition to energy and bandwidth usage concerns. The sensed event could be mobile as well, i.e. target tracking. Sensing a fixed event allows the network to operate in a reactive mode, while dynamic events mostly use periodic reporting to the sink as a means of communication.

- *Operating Environment:* Sensor nodes and their networks are more likely to be placed in a more hazardous environment as opposed to ad-hoc nodes and networks. Depending on the environment, there may be concerns with regards to construction, component and sensing tolerance levels.

[2] Available from www.xbow.com

- *Transmission Media:* The medium for transmission in a WSN is, as stated wireless, and is usually low in bandwidth (1-100 kbps). Some of the problems a wireless channel will suffer from include high error rates and fading. WSNs make use of two methods in sensor node MAC layer communications, namely TDMA and CSMA. TDMA conserves more energy than a CSMA approach but has higher setup requirements. WSNs are more prone to various wireless communication phenomenon, such as propagation and fading. Most of these are attributable to the limited transceivers employed due to power constraints. Many WSNs will also be placed in awkward positions where environmental factors will seriously impact the transmission and reception of any signal.

- *Energy Consumption:* This is possibly the most important design challenge when it comes to WSNs. Nodes consume their energy by transmitting, processing and computing messages. Sensor node lifetime is strongly dependent upon battery lifetime [9]. The energy consumed during the transmission of messages can be up to 75% of the available energy resource [10]. Sensor nodes usually conduct two tasks in a WSN, that of a data sender and a data router. The functions of a sensor node can be broken up into communication, sensing, and processing.

  - *Communication:* As mentioned before, the communication of the messages is known as the greatest consumer of energy. The transceiver consumes energy during wake-up and active states. The size of the message being sent, consisting of the data and all the headers, plays an important role in the amount that is consumed. Smaller messages are obviously better for a WSN. A cross-layer approach to the design of a node and its protocols should be able to decrease the number of bits contained in the headers. The MAC layer is responsible for deciding on the state of the transciever, either awake or asleep. The power consumption as shown in [5] and calculated in [11] is as follows:

$$P_C = N_T[P_T(T_{on} + T_{st}) + P_{out}(T_{on})] + N_R[P_R(R_{on} + R_{st})] \tag{2.3}$$

    where N_T is the number of times the transmitter is switched on per time unit, P_T is the transmitter power consumption, T_{on} is the transmitter on time, T_{st} is the transmitter start-up time, P_{out} is the transmitter output power, N_R is the number of times the receiver unit is switch on per time unit, R_{on} is the receiver on time and R_{st} is the receiver start-up time.

  - *Sensing or Acting:* Depending on the sensor or actuator that is being employed by the node, this function could consume more energy than communications, although this is usually limited to only actuators. Most application networks will only make use of sensor nodes. The sensing function is generally considered to be the second largest consumer of the energy resource.
  - *Processing:* The function that consumes the least amount of energy available to a node, processing involves the calculations and management for MAC, routing and application layers. To keep the consumption to a minimum, the calculations should be kept as short and simple as possible.

- *Fault Tolerance:* Nodes will fail, this is a given. How the network and algorithms react to these failures is a design concern. A failure of a single node should not affect the overall task of the sensor network. Fault tolerance is mainly handled by the MAC and routing layers. Some of the options available to these layers may include adjusting transmitting powers and signalling rates on existing links. Multiple levels of redundancy may be called for in designing of these layers.

- *Node Deployment:* The three phases for topological changes and maintenance are the *deployment, post-deployment and re-deployment phases*. Initial setup takes place during the deployment phase, topology changes occur during the post-deployment phase and new nodes are added to or moved in the network in the re-deployment phase. Node deployment is very application dependent. Nodes can be dropped from an aeroplane (randomized) for example, or placed manually (deterministic). The connectivity and topology of the network can be defined beforehand by the manual placement of the nodes, thus allowing data to travel along predetermined paths. A random displacement of nodes brings a more ad-hoc scheme to the routing infrastructure. Post-deployment topology changes can occur due to [4]:

  - position,
  - reachability,
  - available energy,
  - malfunctioning, and
  - task details.

Sensor nodes can be redeployed to replace nodes that have had a failure, or to increase the node density to improve sensor accuracy. These added nodes or redeployed nodes will lead to new routing paths and decisions being established and required.

- *Connectivity:* Each node in a network effectively has a connection with any neighbours that it can communicate with. A high node density precludes the nodes from being isolated in the network. These possible connections are still dependent on the variable nature of the WSN topology and shrinkage due to node failures. Lastly, the connectivity is dependant on the possibly random distrubution of the nodes in the network.

- *Coverage:* Each sensor in a WSN has a certain view of its surrounding environment. This view is limited in both range and accuracy. Depending on the application, the required node density, and the fact that the sensor can only cover a limited physical area, the area coverage becomes an important design parameter.

- *Data Aggregation:* Sensor nodes may generate large amounts of redundant data, e.g. similar data packets from multiple nodes. This redundancy can be shrunk through aggregation of the data as it travels through the network. Aggregation is the process of combining data from different sources according to certain functions, e.g. duplicate suppresion, minima, maxima, average. Signal processing methods can also be used, this process being known as *data fusion*. An example of this technique would be beamforming, which combines incoming signals and reduces the noise present.

2.2.4 Wireless Sensor / Actor Networks

A new area of research in the WSN field, WSAN has unique characteristics additional to the traditional ones discussed [12]. A WSAN is not only populated by sensor nodes but also by actor nodes. These nodes differ from the traditional sensor nodes, by having actuators that allow them to interact with the physical world. These characteristics are:

- *Real-time requirement:* Certain WSAN applications may require a real-time response to some sensed information. A sensor may detect a fire, an extreme rise in temperature, and transmit this information to an actor that is in control of the fire suppression system. The actor is required to act upon the information in a timely fashion, i.e. before the fire becomes uncontrollable.

- *Coordination:* In a WSN environment a central node, the sink node, handles data collection and coordination. This differs in a WSAN in that the actor nodes themselves handle data collection from surrounding sensor nodes and coordination as well. Coordindation can take place in two fashions, either sensor-actor or actor-actor. An example of this may be sensors providing data to an actor regarding an event, and the actor nodes coordinate in the network deciding how best to act upon the information.

The role of the nodes in a WSAN is to collect the data and perform actions based upon this information. An example of this can be seen in *Fig.* 2.3. The sensor and actor nodes are deployed in the *sensor/actor field* while the sink node monitors the network. The sink node may also be communicating with a "task manager node" to ensure optimum performance in the network. Sensed information can flow in two ways to the actor nodes, either to the actors themselves, or via the sink node, which then sends the information to the actor nodes. These types of architectures are known as either automated or semi-automated respectively, as shown in *Fig.* 2.4. Protocol and algorithms for WSANs have unique objectives, as discussed in [12], that are not necessarily found in a WSN. These objectives are to:

- *provide real-time services with given delay bounds, according to application constraints,*
- *ensure energy efficient communication among sensors and actors,*
- *ensure ordering between the different events when they are reported to the actors,*
- *provide synchronization among different sensors reporting the same event to multiple or same actor in order to facilitate a one-time response in the entire region,*
- *track and report the sensed phenomena to a different set of actors not necessarily based on proximity or energy limitations for the case when the events take place in different locations.*

2.2.5 Wireless Communication Standards

Wireless sensor networks usually use low-data rates and short ranges for communication. Some of the more common standards are discussed briefly. Some of the standards used in WSNs are shown in *Fig.* 2.5.

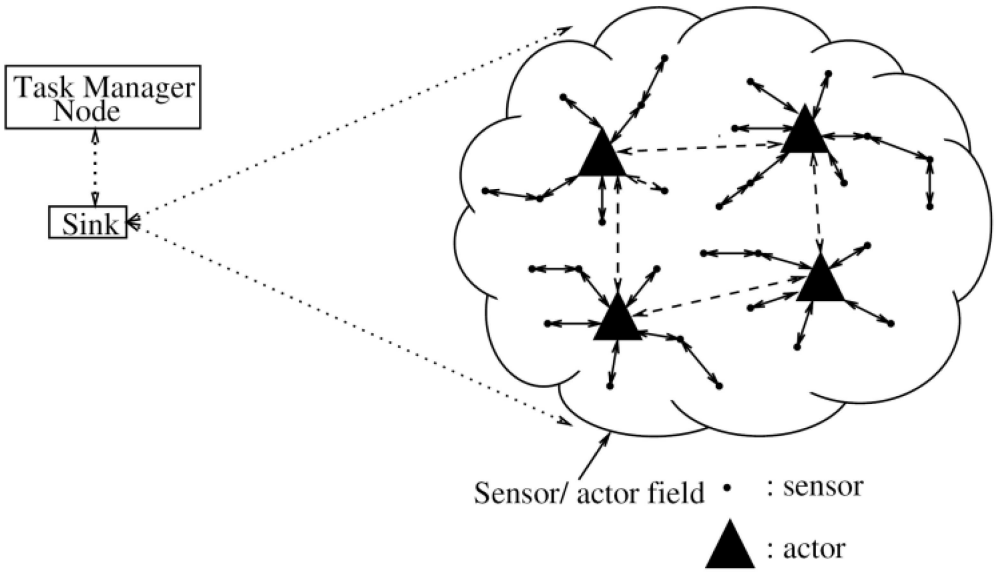

FIGURE 2.3: The physical architecture of WSANs. Taken from [12] (Fig. 1.).

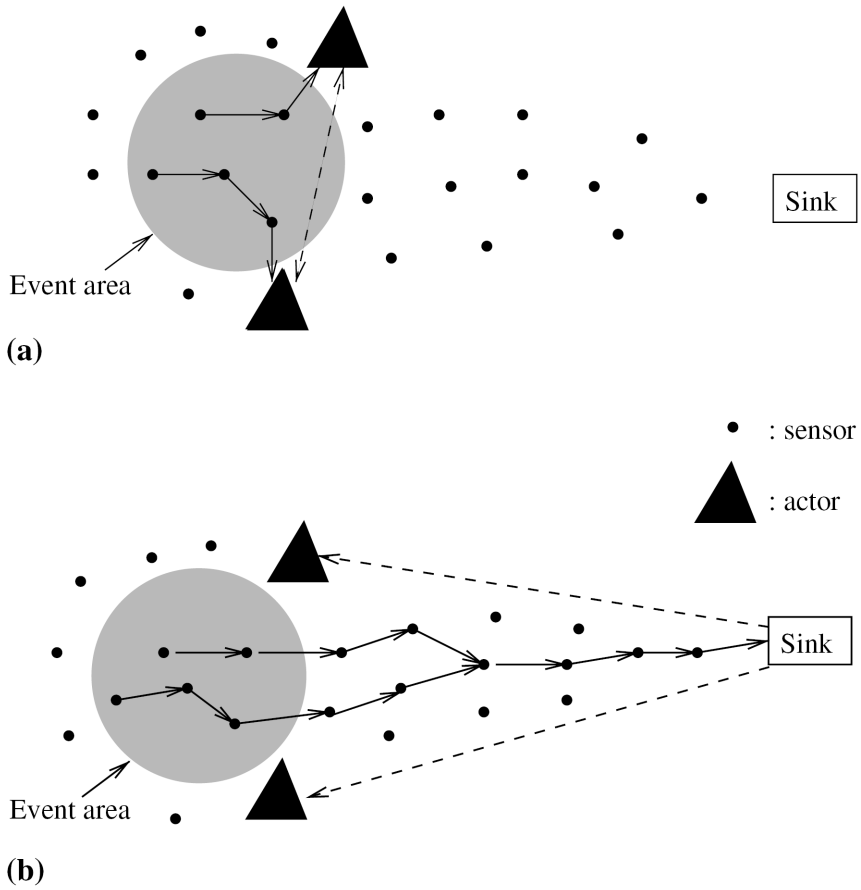

FIGURE 2.4: (a) Automated vs. (b) Semi-Automated Architecture. Taken from [12] (Fig. 2.).

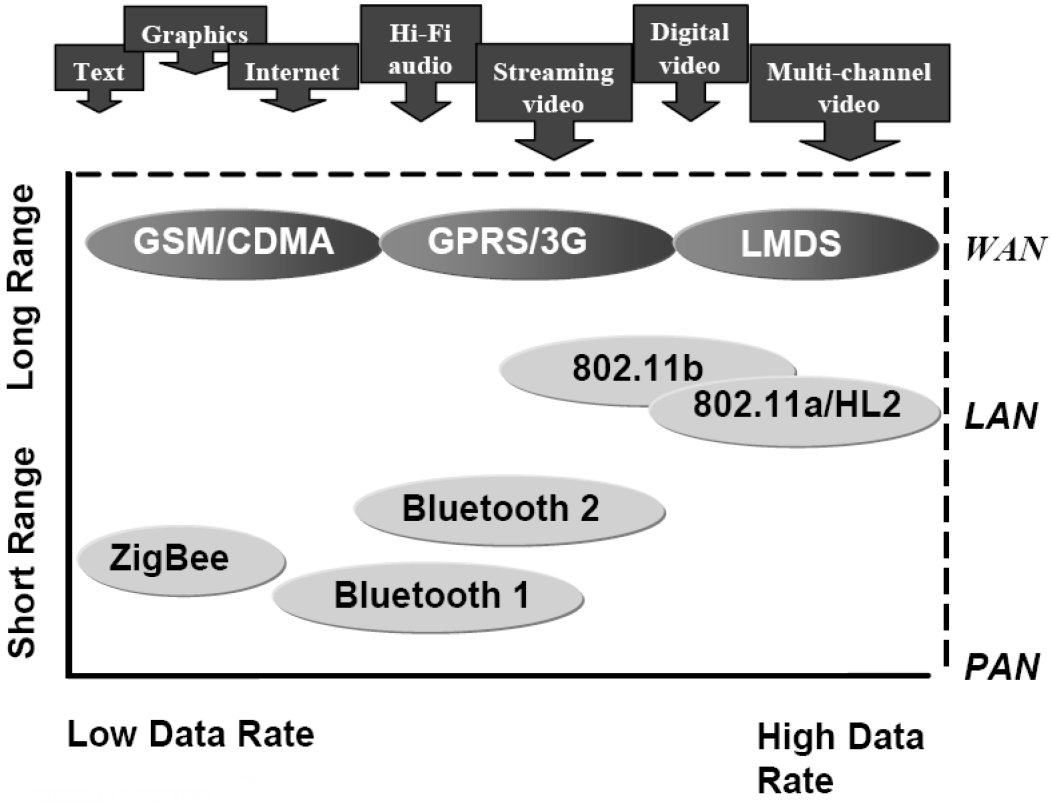

FIGURE 2.5: Wireless Communication Standards. Taken from [13].

2.2.5.1 Bluetooth

Bluetooth is a wireless protocol that has been around since the late nineties. Bluetooth, like some of the other protocols described here, also operates in the free 2.4 GHz ISM band. Bluetooth is more suited for low speed applications such as voice or data. Many products are seeing their wires cut and replaced with Bluetooth connections as shown in *Fig.* 2.6. Bluetooth is also one of the more predominate technologies used in cellular phones.

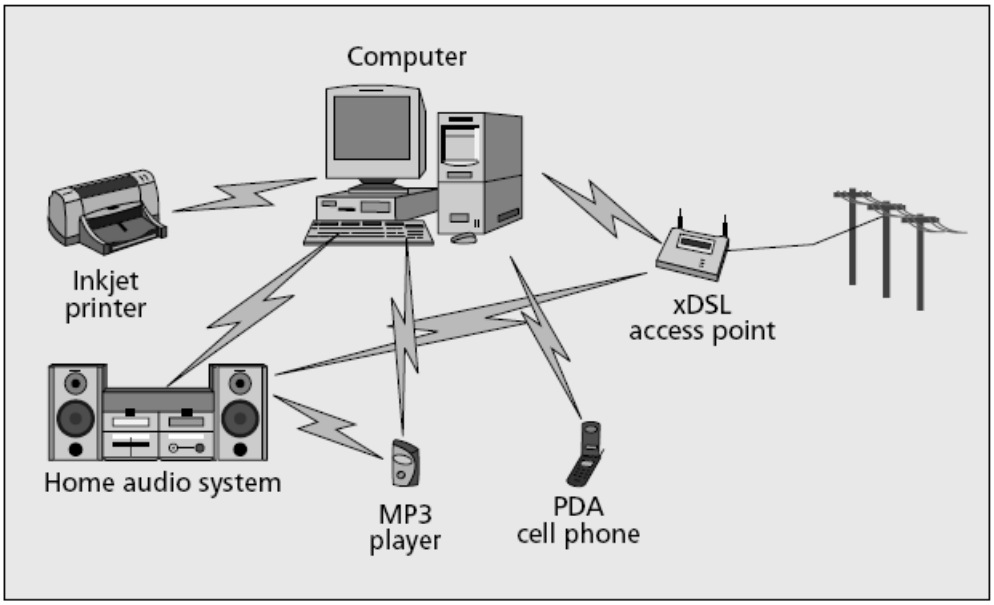

FIGURE 2.6: Examples of Bluetooth applications. Taken from [14] (Fig. 1.).

Bluetooth offers a maximum transmission speed of 1 Mbps [15] over a range of 10m (class 2 device). If a power amplifier is introduced, this range can be extended to 100m (class 1 device). Bluetooth uses FHSS in its communications, which means basically that it takes the available spectrum and breaks it up into sections, in this case 79 hops. This allows for better transmission protection against interference noise on any one channel as the data packets can be hopped to other frequencies where the interference is less.

Bluetooth Network Topology

Bluetooth networks can be set up easily, sometimes all that is required is to enter the range of another Bluetooth device. The ad-hoc networks can be point-to-point or point-to-multipoint. The requirement to make sense from what could end in chaos, is that one of the Bluetooth

devices must become a master which controls the network transfers, whereas the other devices will become slaves. With current specifications, up to seven slaves can communicate with a single master. Bluetooth networks are sometimes more commonly referred to as piconets , an example is shown in *Fig.* 2.7.

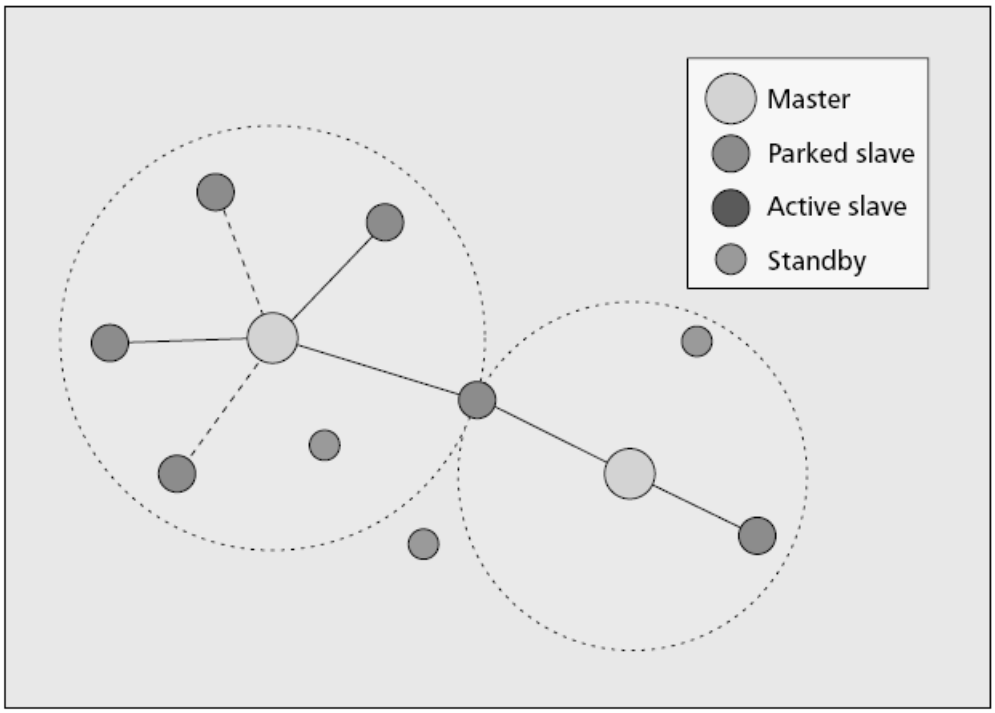

FIGURE 2.7: Example diagram of what a Bluetooth piconet may look like. Taken from [14] (Fig. 2.).

2.2.5.2 ZigBee

ZigBee is a new standard that has been ratified by the IEEE , specification 802.15.4, and is defined in [16]. ZigBee was designed for low-rate PAN's in the industrial or home markets. One of the advantages designed into ZigBee is that it has a low-duty cycle, i.e. spends most of its time snoozing. The idea behind this is that a node on a network could run for months and perhaps even years on standard alkaline batteries, only waking up when information needs to be transmitted, received or acted upon.

There are many standards that provide for fast data transfer rates, i.e. UWB (described later), that cater for services such as video or PC LAN's. The need existed for a low-power, low-rate wireless standard, hence ZigBee. By using the PHY-layer, MAC-layer and ZigBee's

Network and Application Support Layer the following features are provided:

- extremely low cost,
- ease of implementation,
- reliable data transfer,
- short range operation,
- very low power consumption, and
- appropriate levels of security.

ZigBee can operate in the 868 MHz (20 kbps), 915 MHz (40 kbps) and the ever popular 2.4 GHz (250kpbs) ISM band thus joining Bluetooth and WiFi. Theoretically operational ranges extend from 0 metres to 75 metres and perhaps further, depending on environmental conditions. One of the advantages of ZigBee over Bluetooth is the nature in which the protocols work. Memory sizes required for the ZigBee stack range from 3 Kb to 32 Kb, this can compared to the average Bluetooth requirement of 250 Kb. ZigBee uses DSSS to transmit data over the channel; representation of this can be seen in *Fig.* 2.8. DSSS has proven to be a reliable and efficient way to transmit data and provides sufficient protection from interference.

ZigBee also has the support of numerous companies, these are listed on the website of the ZigBee Alliance (www.zigbee.org).

ZigBee Network Topology

ZigBee uses a simple master-slave principle. A master node is required to manage the network and to determine network traffic needs. Each master node can support up to 254 slave nodes. ZigBee can in fact use either 16-bit addressing or IEEE 64-bit addressing schemes. By interconnecting master nodes it is possible to expand the network even further. Another advantage is that when a node is placed in a low-power sleep mode it becomes possible for the node to wake up and transmit a packet in 15 milli-seconds as mentioned in [18]. *Fig.* 2.9 shows typical ZigBee network topologies.

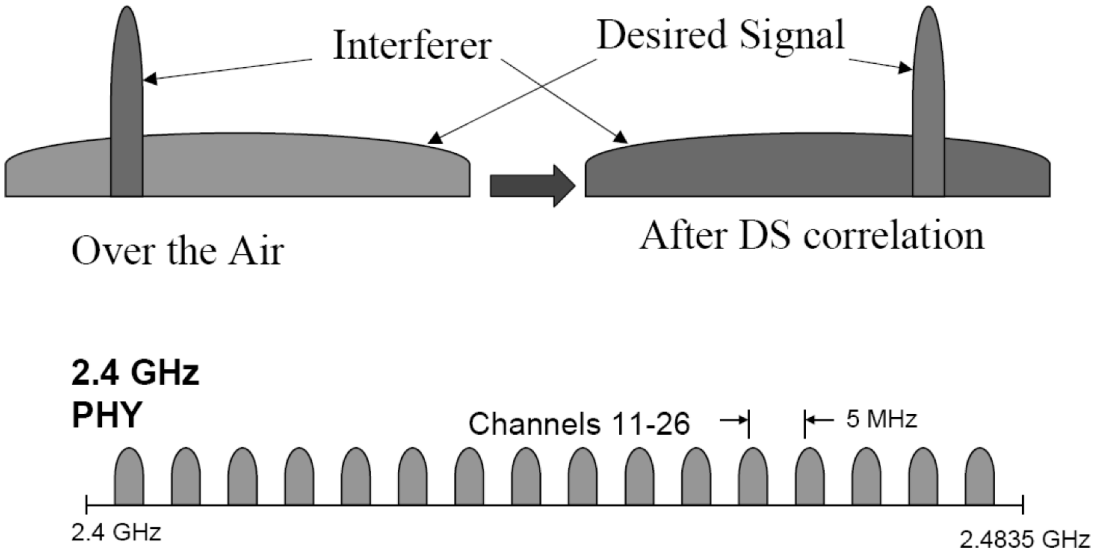

FIGURE 2.8: Illustration of ZigBee DSSS and a 2.4 GHz band representation. Taken from [17].

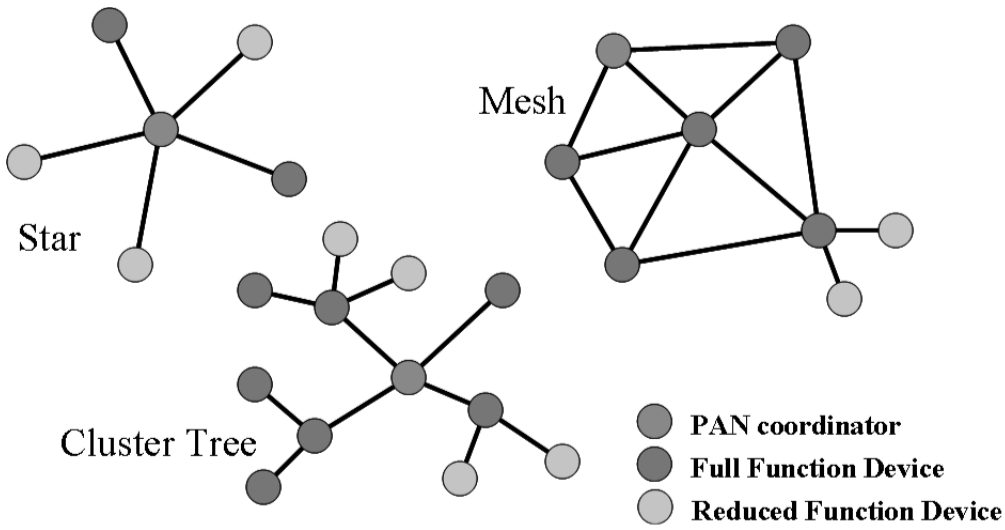

FIGURE 2.9: Typical networks topologies with ZigBee. Taken from [19].

ZigBee defines two types of devices; full function devices (FFD) and reduced function devices (RFD). Their functions are presented in *Table* 2.1.

Full function device	Reduced function device
Can function in any topology	Limited to star topology
Capable of being the network coordinator	Cannot become a network coordinator
Capable of being a coordinator	Talks only to a network coordinator
Can talk to any other device	Very simple implementation

TABLE 2.1: Functions and capabilities of ZigBee node devices. Taken from [20].

ZigBee might be a relatively new standard but it is projected [18] that up to fifty nodes may be present in residential homes within four to five years. This wireless standard fits into a segment of the market that has only recently become economically viable.

2.2.5.3 WiFi

This standard describes the creation of what is known as a WLAN. The WiFi standard, also known as 802.11, defined in [21] is made up of various forms, these are depicted in *Table* 2.2.

	IEEE 802.11	IEEE 802.11b	IEEE 802.11a	IEEE 802.11g
Ratification	June 1997	Sept. 1999	Sept. 1999	June 2003
RF band	2.4 GHz	2.4 GHz	5 GHz	2.4 GHz
Max. data rate	2 Mbps	11 Mbps	54 Mbps	54 Mbps
Physical layer	FHSS, DSSS, IR	DSSS / CCK	OFDM	OFDM
Typical range	50-100m	50-100m	50-100m	50-100m
Ratification	June 1997	Sept. 1999	Sept. 1999	June 2003

Table 2.2: Specifications of the various types of WiFi (802.11). Adapted from [22] (Table. 1.).

As can be seen, 802.11 has different data rates for each of its instances. The most popular standards commercially are 802.11b and 802.11g. These products might not be as fast as common cable networks but can prove more efficient and cost effective in some

implementations. Most of WiFi occupies the 2.4 GHz ISM band and 802.11a situated in the 5 GHz band. Unfortunately the 5 GHz band is not freely available in all countries and this is hampering its market potential. WiFi has the advantage over some other wireless standards in being able to cover a footprint extending to 100m with high data rates. Various radio techniques are used on the physical layer of WiFi. The most important ones being DSSS, and FHSS.

WiFi network topology

A BSS defines two types of networks; ad-hoc and infrastructure. Similar to Bluetooth, ad-hoc networks are created as devices enter the wireless footprint. An ad-hoc network has stations that communicate with each other, and no access points to any other networks. This is sometimes referred to as an IBSS . Infrastructure networks use access-points (AP) to provide access to other networks. This is similar to the function that a gateway performs in modern cable networks. *Fig.* 2.10 depicts these two types of networks.

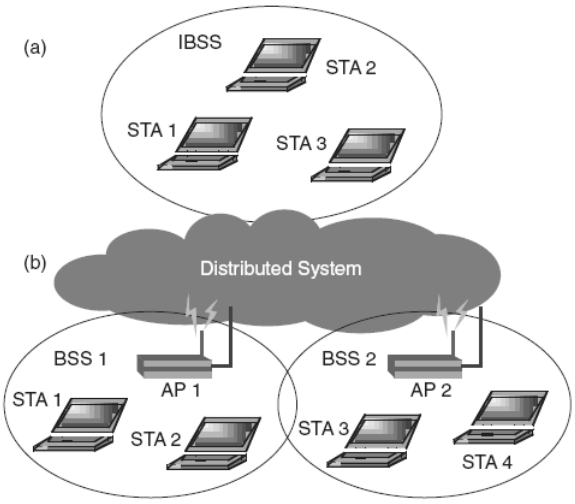

FIGURE 2.10: a) An ad-hoc network showing stations (STA). b) An infrastructure network showing stations and access points (AP). Taken from [22] (Fig. 1.).

2.2.5.4 Ultrawideband

A technology that has recently been standardized, IEEE 802.15.3a, for use in personal area networks, but was first discovered as early as the 1800s by Hertz and Marconi [23]. However in 1910 attention was focued onto narrowband communications. This led to a lull in research,

but in the 1960s UWB made a come back in military and police appilcations, e.g. radar. Ultrawideband could be considered the opposite of ZigBee, as it has a shorter range (< 10 metres) and allows higher transmission speeds (480 Mbps at 1 metre [24]). Ultrawideband operates in the unlicensed 3.1-10.6 GHz range and transmits information by spreading it over a large bandwidth (> 500 MHz). Traditionally thought of as a type of "pulse radio", the FCC and ITU-R now define UWB in terms of a: *transmission from an antenna for which the emitted signal bandwidth exceeds the lesser of 500 MHz or 20% of the center frequency* [25]. Basicallly UWB does not use a carrier frequency, like other radio systems. The pulses are used to transmit the information in a similar format as a digital signal. For comparison see Table 2.3.

Wireless Technology	Highest Speed (Mbps)	Frequency Range (GHz)	Cell Radius (m)
Wibree	1	2.4	5-10
UWB	480	3.1-10.6	10
Bluetooth	1	2.4	10
802.11b	11	2.4	50
802.11a	54	5	30

TABLE 2.3: Comparison of WSN standards.

2.2.5.5 Wibree

A new standard designed by the Nokia corporation and recently released as an open industry standard initiative. Designed as a low-power low-bandwidth standard that could be used on low-power devices. An example would be a watch that is able to communicate the time to another device, or perhaps the watch is able to receive synchronization updates keeping the time synchronized down to the second. The claim is made that Wibree is the first wireless technology that achieves the following, as listed in [26]:

- *Ultra low peak, average and idle mode power consumption*
- *Ultra low cost and small size for accessories and human interface devices (HID)*
- *Minimal cost and size addition to mobile phones and PCs*
- *Global, intuitive and secure multi-vendor interoperability*

The Wibree specification lists two types of modes, namely dual-mode and stand-alone. Dual-mode implementation involves the radio circuitry being shared with the Bluetooth radio, a sort of add-on. Dual-mode was designed for mobile phones, multimedia computers and PCs. Example applications for the stand-alone implementation are any power and cost optimized designs, e.g. human HID product catergories. Wibree has a unique feature, all of its traffic can be scheduled in between those of Bluetooth. Some detailed specifications include:

Radio Specification: Wibree is designed for ultra low-power consumption. Wibree operates in the 2.4 GHz ISM band with physical layer bit rate of 1 Mbps and provides a link distance of 5-10 metres.

Link-layer Specification: Wibree's link layer provides a number of features; ultra low-power idle mode operation, simple device discovery and reliable point-to-multipoint data transfer with advanced power-save and encryption capabilities.

2.3 CHAPTER SUMMARY

This chapter introduced the basic concepts of the WSN environment. By looking into the constituent parts of a sensor node it gives us an idea of the limited resources that are availble to the node. The difference between the WSN and Ad-Hoc networks were also covered. It is important to realise that the two networks occupy different areas and applications, but that there is minimal overlap.

Any WSN will generally comprise of a multi-hop routing scheme. This and other routing challenges were discussed and were brought to the fore, as they should be considered when designing any new routing protocols or algorithms. The largest user of the available energy reserves is communication. It was pointed out that any transmissions should be curtailed wherever possible.

The differences and unique requirements between a WSN and WSAN were briefly discussed, and it was pointed out that WSANs introduce interesting routing situations, namely

automated or semi-automated in nature.

Lastly, the possible wireless communication standards that are used in the WSN environment were listed and a brief introduction given. The two main contenders in the future will most likely be those of ZigBee and UWB. Each of these two have their set of applications that they would excel in, depending on the required bandwidth.

CHAPTER THREE
ROUTING PROTOCOLS

3.1 CHAPTER OVERVIEW

A summary of the protocols that were researched during the literature study can be found in this chapter. Designing for cross-layering, simulators and their credibility are covered towards the end of the chapter.

3.2 WIRELESS SENSOR ROUTING PROTOCOLS

3.2.1 Routing Protocol Classifications

Routing protocols can usually be classified by three main criteria; network type, communications initiator and method used for path establishment. The network types are listed below:

- *Direct:* The earlist form of protocol for a WSN, although not really a routing protocol as such, as no information is sent via any other nodes. Should a message need to be sent, the node will communicate with the sink node directly. The crux with a protocol of this type is the fact that the expandability and network size becomes an issue. It is crucial that nodes be placed where they are able to communicate with the sink node.
- *Flat:* Protocols of this nature allow for the messages to be routed via other nodes in their quest to reach the sink node. The unique element is that all the nodes have the same information regarding the state of the network, which varies from protocol to protocol.
- *Hierarchical:* These types break the network into groups of nodes, known as clusters. Each cluster usually has a single node that is responsible for communication control in the cluster. The normal nodes transmit their messages to the clusterhead and

the clusterhead will forward the messages to the sink. This was originally done using a high-power transmitter, but some protocols have the clusterheads forward the messages through other clusterheads, routing or normal nodes.

The communications initiator is an important aspect in wireless networks as it has an impact on the energy efficiency. These are explained as follows:

- *Source:* In this situation the source of the message is the initiator. The node that has witnessed an event, event-driven or time-driven, will create the message and transmit it to the sink. This method usually proves the most energy efficient.
- *Destination:* Usually the sink node, the destination node propagates a message into the network looking for specific information, query-driven. If the sink node does not know the location of the node required, the message is broadcasted. This introduces significant overhead into the environment.

For messages to be transmitted effectively to the sink node, a path has to be established. A path may be established in three ways:

- *Proactive:* Similar to routing protocols used on the Internet's infrastructure, nodes build routes before they are needed and these are stored in some type of routing table. The problem with this method is that a path can become unavailable. This change in network topology has to be propogated through the network. This means more messages, which in turn suggests greater power consumption. With each path having a slot in the routing table and possible network sizes in the thousands, it is not realistic to have sufficient resources (memory) for routing tables of this size.
- *Reactive:* Should a node have information to pass on, the node will create the path on which to send the message. In other words the routing paths are created as is needed.
- *Hybrid:* A combination of the two, proactive and reactive.

These classification criteria can be seen in *Fig.* 3.1.

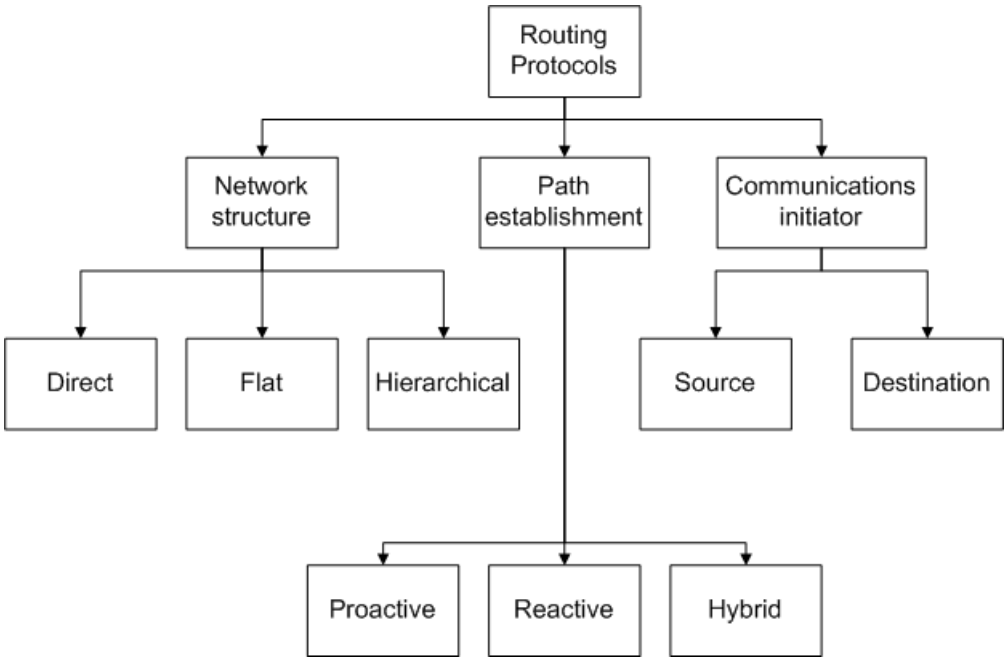

FIGURE 3.1: Routing Protocol Classifications.

3.2.2 Flat Architecture

3.2.2.1 Flooding and Gossiping

Two of the original protocols designed for WSNs, Flooding and Gossiping [27] do not require any routing algorithms or topology maintenance. In Flooding, each sensor node will broadcast its information. Each receiving node will then pass the message on, until the message reaches the sink node or the TTL value is exceeded. Gossiping is an altered version of Flooding. The sending node will select a neighbour at random to send its data to. This way the information travels around the network with the hopes that the message will reach the sink at some point, after a possible delay. Some of the problems that are faced by these protocols are *implosion Fig.* 3.2 and *overlap Fig.* 3.3. Implosion is caused by duplicate messages being sent to the same node, and overlap deals with two nodes sensing the same region and both reporting their values. By sending its data to one random neighbour Gossiping is immune to implosion. Unfortunately that is not the end of the problems for these two protocols. Both of these protocol suffer from *resource blindness*, a term used to refer to the rampant use of resources (energy) but with no regard to the amount consumed [28].

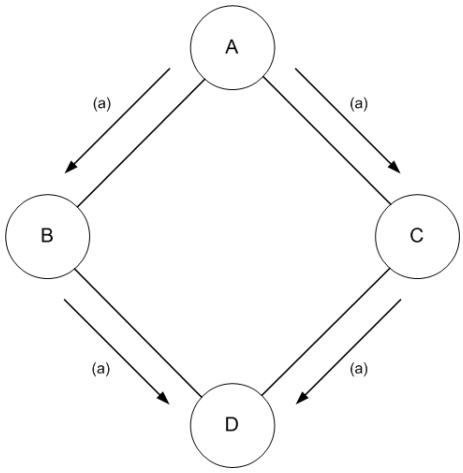

FIGURE 3.2: *Implosion:* Node A begins by flooding its data to all of its neighbours. D gets two copies of the same data eventually, which is unnecessary. Redrawn from [28] (Fig. 1.).

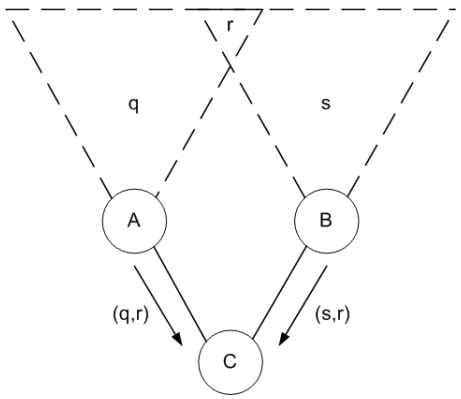

FIGURE 3.3: *Overlap:* Two sensors cover an overlapping region ("r") and C gets the same data from A and B. Redrawn from [28] (Fig. 2.).

3.2.2.2 Sensor Protocols for Information via Negotiation

Sensor Protocols for Information via Negotion [28] (SPIN) is another example of the early work that was done with regards to WSNs. Information in SPIN is first described using a form of meta-data. This meta-data is exchanged before transmission using an advertisement (ADV) mechanism. The neighbours that are then interested in this data send a request message (REQ) to the originator of the ADV message. The advantage of this process is that implosion, overlap and resource blindness are not a factor, giving SPIN a significant energy saving (up to a factor of 3.5). This entire process can be seen in *Fig.* 3.4.

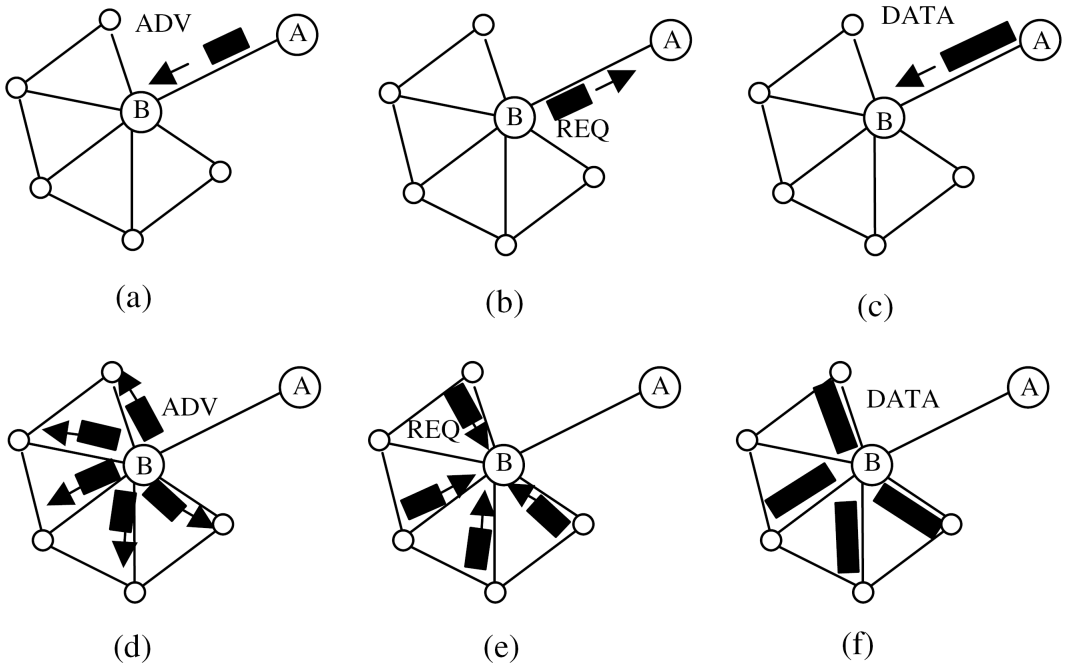

FIGURE 3.4: SPIN protocol. Node A starts by advertising its data to node B (a). Node B responds by sending a request to node A (b). After receiving the requested data (c), node B then sends out advertisements to its neighbours (d), who in turn send requests back to B (e,f). Redrawn from [28] (Fig. 3.).

Topological changes are limited to immediate neighbours only, as this is the limit of the nodes' awareness of the network. If all the neighbours are interested in the data advertised then SPIN generates a number of messages, but this is application dependent. A problem occurs if no one is interested, then the sink would never receive any information from that node.

3.2.2.3 Directed Diffusion

Directed Diffusion [29] is recognised as being an important milestone for routing in WSNs. Many other protocols are built on its foundation [30]. An interest for specific data is "diffused" through the network, where a naming scheme is used for the data.

For receiving data messages the Directed Diffusion protocol is divided into three phases; interest propagation, initial gradients setup, and data delivery. This process is summarized in *Fig.* 3.5.

- *Interest propagation:* The interest, defined using a list of attribute-values pairs, is broadcast by the sink node. Caching of the interest can be done by the receiving node for later use. Each node maintains a interest table, where all received interest messages are cached. The interest is then compared to the data received from other sensor nodes.
- *Initial gradient setup:* Directed Diffusion makes use of gradient values (data value, duration and expiration time) that are located in the interest message to establish paths between the sink node and the sources of the data. Several paths can be established, but one will be reinforced by the sink sending the interest (with a lower interval) again. Thie effectively increases the delivery of data to the sink.
- *Data delivery:* Data aggregation is performed by the nodes, thereby increasing energy efficiency. A sensor node will generate the traffic at the required rate, and will transmit this data to the sink via the established path. The duration and expiration values received from the interest will control the flow of traffic from the sensor node.

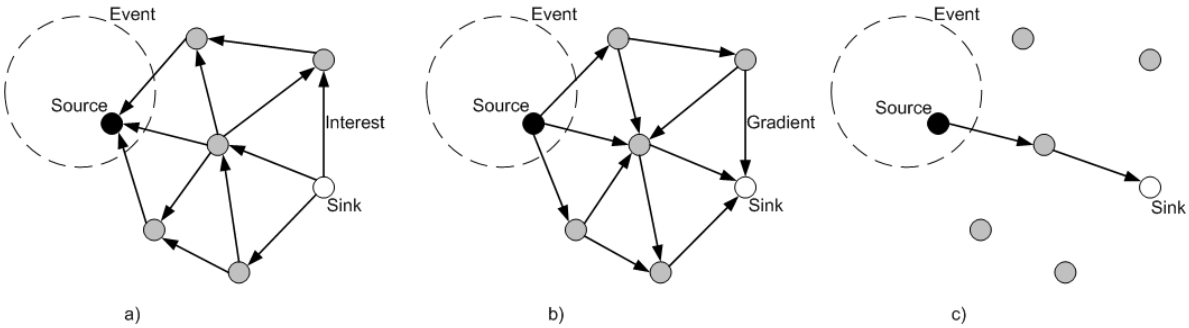

FIGURE 3.5: Directed Diffusion protocol phases. (a) Interest propagation, (b) initial gradients setup, (c) data delivery along reinforced path. Redrawn from [29] (Fig. 1.).

With Directed Diffusion employing a query-driven model, applications requiring continuous data delivery will not work efficiently. The afore mentioned naming schemes are application dependent, making the protocol unique to each and every implementation.

Energy efficiency is not at an optimal level as the interests and gradients are flooded onto the network and the memory required to maintain the interest tables for a large network is substantial. Thus Directed Diffusion can consume large amounts of energy if the application is not ideally suited to its operation.

3.2.2.4 Energy-Aware Routing

Energy-Aware Routing [31] is designed to choose sub-optimal paths using a probability function, which depends on the energy consumption of each path. By doing this, the hope is that the network lifetime will be extended to its fullest. One assumption that the protocol places on the overall network is that the nodes themselves are addressable via a class-based addressing scheme, which includes the location and type of the node. The three phases of the protocol are briefly outlined, as they appear in [31].

1. *Setup phase: Localized flooding occurs to find the routes and create the routing tables. While doing this, the total energy cost is calculated in each node. For instance, if the request is sent from node N_i to node N_j, N_j calculates the cost of the path as follows:*

$$C_{N_j,N_i} = Cost(N_i) + Metric(N_j, N_i) \tag{3.1}$$

Here, the energy metric used captures transmission and reception costs along with the residual energy of the nodes. Paths that have a very high cost are discarded. The node selection is done according to closeness to the destination. The node assigns a probability to each of its neighbors in routing (forwarding) table (FT) corresponding to the formed paths. The probability is inversely proportional to the cost, that is:

$$P_{N_j,N_i} = \frac{1/C_{N_jNi}}{\sum_{k \in FT_j} 1/C_{N_jN_k}} \tag{3.2}$$

N_j then calculates the average cost for reaching the destination using the neighbors in the forwarding table (FT_j) using the formula:

$$Cost(N_j) = \sum_{i \in FT_j} P_{N_j,N_i} C_{N_j,N_i} \tag{3.3}$$

2. *Data communication phase: Each node forwards the packet by randomly choosing a node from its forwarding table using the probabilities.*

3. *Route maintenance phase: Localized flooding is performed infrequently to keep all the paths alive.*

The problem with this protocol is two-fold. Firstly, the protocol assumes that nodes are aware of their location and that there is an addressing scheme being used to address the individual nodes. This complicates the initial set up phase for the network using this protocols. Secondly, only a single path is used for sending information to the sink. By using this method the protocol would struggle to recuperate from a path failure.

3.2.2.5 Simple Energy Efficient Routing

Simple Energy Efficient Routing [5] (SEER) protocol forms the basis for the development of HEER, the protocol designed herein. Performing similarly to Energy-Aware Routing protocol, SEER is a source-initiated protocol that aims to increase node and network lifetimes. The SEER protocol first begins with the sink node(s) sending a broadcast message, a type of “network setup” message, with the hop count set to zero. A sensor node will receive this message and record the node in its neighbour table, the hop count to get there, as well as the current energy level of the sending node. The hop count is then incremented, and senders’ details are replaced with the current nodes’ and the message is then further broadcast. Over time the nodes will have built up a table of all their immediate neighbours. The sink node will periodically rebroadcast its “network setup” message, and the nodes will build up their tables again. Should a sensor node’s energy level reach a certain threshold value, a power (POW) message is broadcast to inform the node’s neighbours that its reserves have reached a critical level.

Once a sensor node has data to transmit, it searches its neighbour table for a node that has the highest available power and the lowest hop count. The message is then transmitted to this neighbour where the process is begun again until the message reaches the sink node. Critical messages are transmitted to two neighbours as opposed to one, the second being the node with the next highest power and lowest hop count.

SEER suffers from the same single path problem as Energy-Aware routing. The path failure would only be corrected once a new message is received from the sink node. The results for SEER supposedly show that its performance is dramatically higher than any of the chosen protocols that it was simulated against. SEER's performance and validity are further discussed in chapter 4.

3.2.2.6 Advantages of a Flat Architecture

Scalability: With each node knowing as much about the network as the next node, this allows the network to be extremely scalable. This is advantageous should the network need to be redeployed or new nodes added. Some protocols are more scalable than others depending on the process of path discovery and the time taken to reach convergence.

Simplicity: From a computational perspective flat protocols are easier to implement than a hierarchical as clusterhead calculations and network setup is kept to a minimum. Protocol operation though can affect the ease with which a flat protocol can be deployed.

3.2.2.7 Disadvantages of a Flat Architecture

Hotspots: Sensor nodes surrounding a sink node, one to two hops away, will consume their energy at a quicker pace. This is due to the amount of messages that have to be routed to the sink. This will eventually lead to the possiblity of a single node handling all the traffic for the network. A way around this would be to increase the number of sink nodes which in turn limits the distance the messages have to travel.

Dis-connectivity: With node failures a prominent feature in WSNs, it is possible for certain sections of the network to become unreachable. If a specific node, located at a critical juncture should fail, the section would be cut off from the rest of the network and unable to reach the sink node.

3.2.3 Hierarchical Architecture

3.2.3.1 Low Energy Adaptive Clustering Hierarchy

Low Energy Adaptive Clustering Hierarchy (LEACH) [9] is one of the more popular hierarchical protocols used today, and was one of the first. Clusters of nodes are formed, and a local clusterhead will route all messages to the sink. Energy is conserved as the nodes only communicate to their clusterheads. The optimal number of clusterheads is estimated at 5% of the overall network size. The clusterheads are selected from nodes selecting a random number, between 0 and 1, that is less than the following calculated threshold value:

$$T(n) = \begin{cases} \frac{p}{1-p*(rmod1/p)} & if n \in G \\ 0 & otherwise \end{cases} \tag{3.4}$$

where p is the desired percentage of clusterheads (e.g. 0.05), r is the current round, and G is the set of nodes that have not been clusterheads in the last $1/p$ rounds. The authors point out the possiblity that a direct protocol may perform better than a minimum-transmission-energy (MTE) protocol. By looking at a simple linear network, as in *Fig.* 3.6, the formula they derive can be seen in *Eq.* (3.5), *Eq.* (3.6), and *Eq.* (3.7).

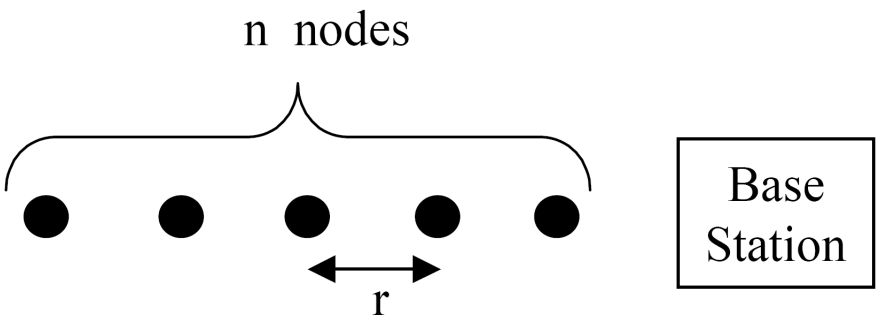

FIGURE 3.6: Simple Linear Network. Taken from [9] (Fig. 2.).

$$E_{direct} < E_{MTE} \tag{3.5}$$

$$E_{elec} + \epsilon_{amp} n^2 r^2 < (2n-1)E_{elec} + \epsilon_{amp} n r^2 \tag{3.6}$$

$$\frac{E_{elec}}{\epsilon_{amp}} > \frac{r^2 n}{2} \tag{3.7}$$

where E_{direct} and E_{MTE} is the energy expended to transmit a message by the two types

of routing protocols. E_{elec} is the energy consumed by the transceiver electronics, ϵ_{amp} the energy consumed by the transmitter amplifier, r the distance between individual nodes, and n the number of nodes.

Once a node has been elected as a clusterhead, it broadcasts an advertisement message (ADV). Sensor nodes decide on which cluster to join based on the received signals strength. The nodes respond to the selected clusterhead using a joining message (JOIN), which informs them of their attachment to the cluster and to the specific clusterhead. The clusterhead is also responsible for distributing TDMA timeslots to all members of the cluster, following which the network is set up. If a data message needs to be transmitted to the sink, the sensor node passes this message on to the clusterhead, which in turn passes it on to the sink node. After a predetermined length of time, the setup phase will be repeated and the clusterheads rotated.

LEACH makes certain assumptions, firstly that each node can reach the sink directly, i.e. should the node be selected as a clusterhead it must be able to communicate with the sink node. The original implementation, as discussed here, is not suitable for large networks, but has been expanded to include multiple hop clusters (LEACH-C).

3.2.3.2 Routing via Energy-Temperature Transformation

Designed to increase the entire networks' lifetime as opposed to a single nodes', Routing via Energy-Temperature Transformation (RETT) [32] is a cluster-based protocol. The protocol is implemented such that the clusterheads are able to select an optimal route to the sink node based on a thermo-dynamic analogy where energy is transformed into temperature and the routing algorithm is searching for the hottest path between source and destination. RETT caters for applications that stipulate that no part of the network is unreachable, due to node failures along critical paths.

Clusterheads are selected based on the temperature values that the nodes currently have. The node with the highest temperature will be selected as the clusterhead. Temperatures are calculated by using the formulas for heat diffusion. When a sensor node has information to send, it transmits the following to the clusterhead:

- *The sensor location.*
- *The sensed data and possibly additional information such as type, value, position, and time.*
- *A timestamp indicating the urgency of the data, which is application dependent.*

After a clusterhead receives this data, it registers the time and copies the data to memory. It then determines whether the information should be sent on immediately, and decides on the best routing path for sending to the sink node. If the data does not have to be sent right away, the node can store the message for a period so that it can wait for more messages to send, a form of data aggregation. The following example, in *Fig.* 3.7, describes the path selection process.

- *Head of cluster I wants to send data to the base station: The selected path becomes I-J-K-L-base station, and the total number of hops are 4.*
- *Head of cluster F wants to send data to the base station: The selected path becomes F-B-C-D-H-L-base station and the total number of hops are 5.*

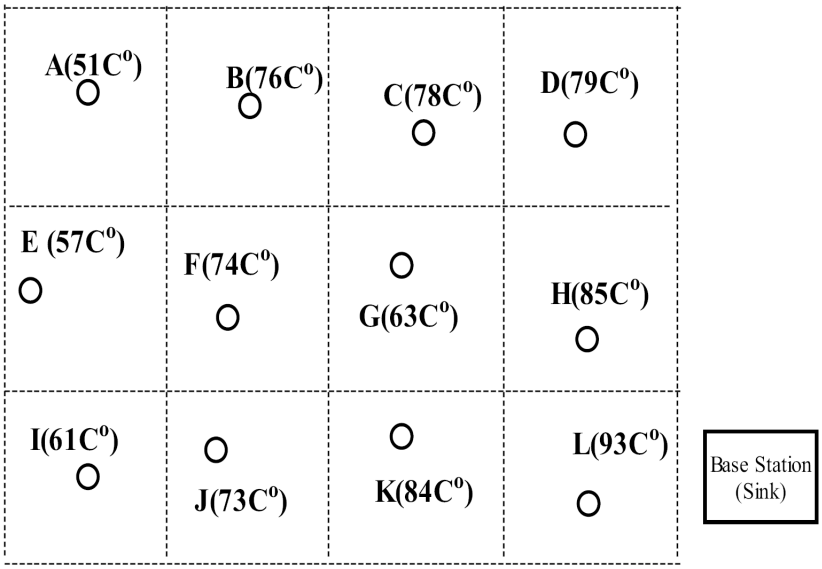

FIGURE 3.7: Selecting the hottest path and/or the shortest path. Taken from [32] (Fig. 3.).

3.2.3.3 Threshold Sensitive Energy Efficient Sensor Network Protocol

The Threshold Sensitive Energy Efficient Sensor Network protocol (TEEN) and the Adaptive Threshold Sensitive Energy Efficient Sensor Network protocol (APTEEN) as proposed in [33, 34] are both hierarchical protocols. TEEN was designed to be responsive, which is important for time critical information. Nodes that are closer together form clusters and this process continues throughout the network until the sink node is reached. An example of this is shown in *Fig.* 3.8.

Once the clusters are formed, the clusterheads transmit a hard and soft threshold to the cluster members. The hard threshold is *the minimum possible value of an attribute to trigger a sensor node to switch on its transmitter and transmit to the clusterhead.* The sensor node will only send its data when the sensed attribute is in the range of interest. Should the value equal or be greater than the hard threshold, the information will still only be sent once the degree of change is greater than the soft threshold value. TEEN does not perform well with periodic data as the threshold value may not be reached, although message traffic can be controlled somewhat by adjusting the threshold values.

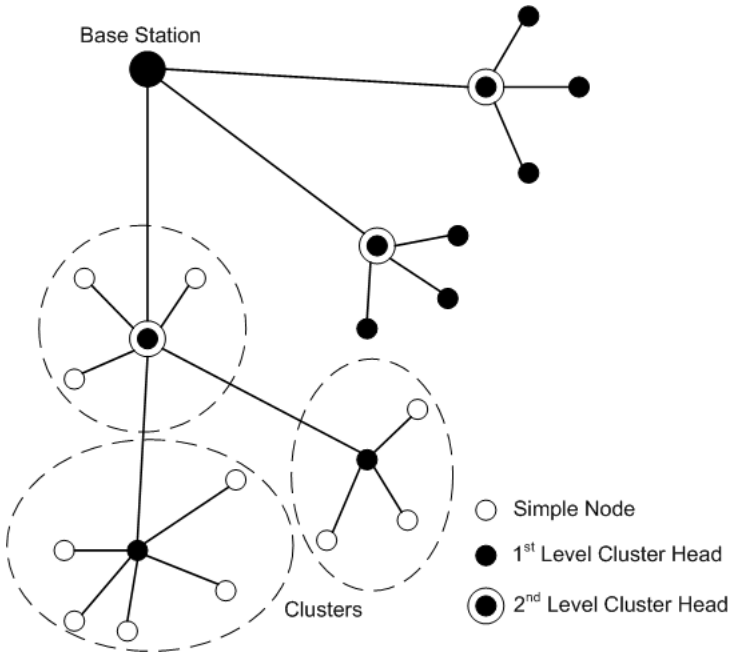

FIGURE 3.8: Hierarchical clustering in TEEN and APTEEN. Redrawn from [33] (Fig. 1.).

APTEEN is an extension of TEEN that allows for better management of periodic data and

time-critical behaviour. The network setup is the same as TEEN, one difference though is that the clusterheads perform a limited amount of data aggregation. APTEEN supports three query types:

1. *Historical: to analyze past data values,*
2. *One-Time: to take a snapshot view of the network,*
3. *Persistant: to monitor an event for a period of time.*

TEEN and APTEEN have been shown to outperform LEACH, with APTEEN being between LEACH and TEEN. The two main drawbacks though are the overheads and complexity associated with cluster formation.

3.2.3.4 Advantages of a Hierarchical Architecture

Data Aggregation: With all the messages for a cluster going through a central location, the clusterhead is able to perform data aggregation on the information before sending the data to the sink.

Localised Power Consumption: The power consumed in a cluster is less than in a whole network, as there is a smaller amount of overhead when setting up the network. Only a small portion of the network (a cluster) is set up, pointing to a clusterhead. Once this has been done, all messages travel a smaller number of hops to reach the clusterhead, thereby saving on their available energy resources.

3.2.3.5 Disadvantages of a Hierarchical Architecture

Hotspots: Clusterheads perform more functions that the average sensor node and this consumes their energy at a greater rate. To alleviate this problem, some protocols rotate the clusterhead amongst all the nodes in the cluster or network. The possiblity of a section getting seperated from the network still exists.

Hardware Requirements: Some protocols require specific hardware, usually a high power transmitter that is capable of reaching the sink node directly. As soon as this happens, the

clusterhead position can no longer be rotated amongst the other nodes, unless of course all the nodes have this facility. As with all features, the cost of the development and production of the nodes will increase.

Complexity: To maintain a hierarchical network is more computationally intensive. The algorithms for clusterhead selection and routing decisions are usually more complex. To provide the initial information for these algorithms to be used and calculated usually requires more knowledge about the network. The only way to learn more is to send and receive more transmissions, increasing the power consumption.

Scalability: Networks that employ nodes with specific hardware requirements decrease their ability to scale to a larger size. As the network would grow so too would the number of clusterheads and they would have to be placed in specific spots so that new clusters could be formed from the additional nodes.

3.2.4 Comparison

A further comparison between the two types of routing architectures can be viewed in Table *Table* 3.1.

3.3 CROSS-LAYER DESIGN

A new idea that has recently become prevalent in WSN research, e.g. [12, 35–38], is the idea of cross-layer design and communication. As opposed to the OSI model where each of the seven layers is treated seperately and the messages are passed between them, the cross-layer approach adds the ability for higher and lower layers to provide information to layers that are not immediate neighbours. This transfer of information allows for greater flexibility and more efficient decisions to be taken, based on more reliable information . An example of this would be if high congestion is detected in the network. Two of the layers, MAC and transport, would then respond:

- *MAC:* by holding off for an exponential back-off [39] period of time before transmitting again.
- *Transport:* by lowering the transmission rates of sensors.

Hierarchical routing	**Flat routing**
Reservation-based scheduling	Contention-based scheduling
Collisions avoided	Collision overhead present
Reduced duty cycle due to periodic sleeping	Variable duty cycle by controlling sleep time of nodes
Data aggregation by clusterhead	Nodes on multihop path aggregates incoming data from neighbours
Simple but not-optimal routing	Routing can be made optimal but with an added complexity
Requires global and local synchronization	Links formed on the fly without synchronization
Overhead of cluster formation throughout the network	Routes formed only in regions that have data for transmission
Lower latency as multiple hop network formed by clusterheads always available	Latency in waking up intermediate nodes and setting up the multipath
Energy dissipation is uniform	Energy dissipation depends on traffic patterns
Energy dissipation cannot be controlled	Energy dissipation adapts to traffic patterns
Fair channel allocation	Fairness not guaranteed

TABLE 3.1: Hierarchical vs Flat topologies routing. Taken from [6] (Table. 2.).

An example of a possible cross-layer design [12] would be when the congestion is high the MAC layer responds by backing-off. If this proves insufficient, the MAC layer could inform the network layer, which could in turn coordinate that data traffic is rerouted through another appropriate node.

The PHY, MAC, routing and signal processing are all related with regards to energy-saving. As a message travels down the OSI model, small headers are added to the data message. In a WSN it is possible for this network information to be larger in size than the actual data message. By combining some functionality between the layers it is possible to minimise the the headers required for network functionality. Experiments with large, high QoS, mobile ad-hoc networks have shown that about 1% of the transmitted bits convey data, while the other 99% support network functionality [36, 37].

Some of the advantages of using the OSI model for wired networks [38, 40] include *taking the long term view, facilitating parallel engineering and ensuring interoperability, lowering development cost and leading to wide implementation.* But contrary to this are also limitations, e.g. *undesired consequences*, to a layered model [38].

3.4 SIMULATORS

3.4.1 NS-2

Developed by the University of California in Berkley as part of the Virtual InterNetwork Testbed (VINT) and funded by DARPA. The two main collaborators in this project were the XEROX Palo Alto Research Center (PARC) and the Lawrence Berkley National Laboratory (LBNL).

This is the most widely used simulator for network research. Many protocols have been written for and can be found for the NS-2 simulator. Popular protocols such as TCP and various routing and multicast protocols can be implemented on a wired, wireless or hybrid network. The simulator is event-driven, coded in C++ and uses Tcl and Object Tcl shells for a visual interface.

The simulator was first used in wired reseach which has since been expanded to include wireless networks, i.e. WiFi. Although not designed for the unique requirements of a WSN, some research has been conducted [41] and an implementation of the ZigBee protocol can be found on the Internet.

Complex networks can be constructed consisting of various network architectures. These networks can compose of any network devices; routers, nodes and connection links. The simulator can also be used in conjunction with the Network Animator (NAM). The NAM allows for the visualisation of the network and packet trace data.

3.4.2 OMNET++

An event-driven simulator, similar to NS-2, allows OMNET++ [42] a certain amount of flexibility. A full version is available for academic purposes, an abundance of documentation, third party code and an active online community via the forums on the website [43] are the main reasons OMNET++ was chosen as the simulator for testing the research in this dissertation.

An OMNET++ model consists of any number of modules, and the depth is unlimited. Modules communicate by passing messages amongst themselves. This feature allows the model to represent any logical structure of any system. Module programming is done via C++ using the included simulation library.

Like NS-2, OMNET++ uses Tcl and Object Tcl to create an interface and visual representation of the simulation. The same interface allows access to module parameters, allowing the user to change these. This is particularly useful during the development and debugging phases. There are a number of third party protocols and simulators that have been programmed on top of the OMNET++ platform, i.e. SENSIM, MAC Simulator, INET Framework, and Mobility Framework (discussed in the next section). Further details regarding the other simulators can be found on the OMNET++ website [43].

3.4.3 Mobility Framework

As mentioned, the Mobility Framework (MF) [44] is built on the OMNET++ platform. It is one of the more complete frameworks, with the most complete documentation, and is a current reference environment for WSN simulations. Node mobility, dynamic connection managment and a wireless channel model are all implemented by the framework. An advantage to using the MF is that *basic modules* are provided that allow a designer/programmer to quickly implement an algorithm or feature.

Each sensor node in the MF consists of a number of modules. These modules are arranged as shown in *Fig.* 3.9. Each layer is responsible for the appropriate actions that occur here, but the noteworthy features are that there are the mobility and blackboard modules. The mobility module is responsible for the position of the node on the actual simulation area. The type of movement, either random or deterministic can be controlled from here. The blackboard is an interesting feature of the architecture as it allows the different layers to communicate directly. Information is published onto the blackboard, and any layer that has subscribed to this data is instantly told about the change and updates their information. This feature allows the cross-layer design of HEER to occur so easily, as it blurs the lines between the layers' communication.

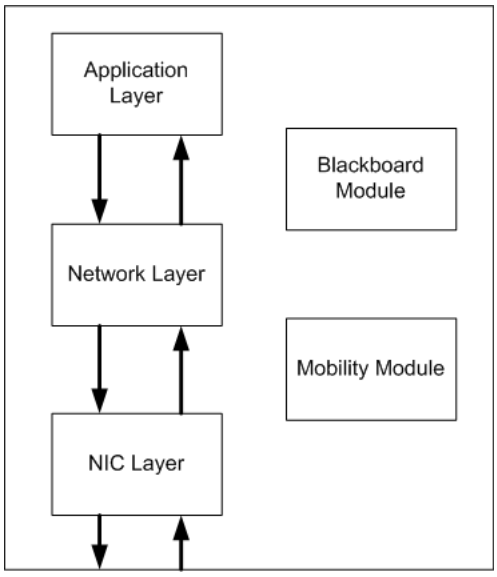

FIGURE 3.9: Structure of a Sensor Node.

The NIC layer module consists of three further modules as can be seen in *Fig.* 3.10. The

snrEval module can be used to calculate information, i.e. SNR, regarding the received signal. The Decider module decides whether the message was actually received, got lost, or has bit errors based on the radio model being used.

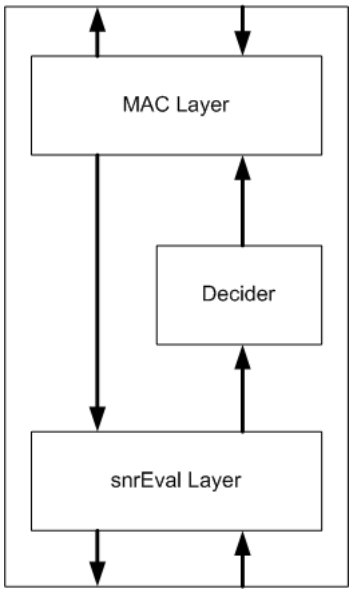

FIGURE 3.10: Structure of the NIC Module

3.5 CHAPTER SUMMARY

To design a routing protocol requires that some basic knowledge be gained from existing research. Routing protocols are classified according to their method of operation, which was briefly discussed in the beginning of the chapter. The protocols that have most impacted on the development of HEER were broken up into either flat or hierarchical designs. These protocols, and their methods and some of their problems were briefly discussed. Lastly, the advantages and disadvantages of the two architectures were listed.

The different types of simulators that are mainly used in WSN research were investigated. The simulator that was chosen, OMNET++ and the Mobility Framework, are further expanded upon in chapter 6

CHAPTER FOUR
CREDIBILITY

"Simulation is useful for evaluating protocol performance and operation. However, the lack of rigor with which its applied threatens the credibility of the published research within the manet research community."

TODD ANDEL AND ALEC YASINSAC

4.1 CHAPTER OVERVIEW

The accuracy or credibility of results is paramount to proving the validity of the chosen research. Without it, the research becomes suspect and essentially null and void in the research community. To this end, this chapter looks into the credibility of simulators, their simulations, and the actual implementations of protocols.

4.2 CREDIBILITY OF WSN SIMULATORS

The motivation regarding the use of simulations in the pursuit of WSN research can be summarised as follows:

- Meaningful analytical evaluation is very difficult.
- The amount of effort required compared to actual physical implementations is low.
- The procedure is very flexible, allowing for quick investigations into alternatives.

By looking at these motivations, it can be seen why simulations are prevalent in WSN research. In some instances it is not feasible to look into actual implementations, i.e. cost involved. Simulations definitely have a place in research but the concern would be whether

they have been implemented correctly. If a simulation is not accurately mimicking reality then the results are meaningless.

As mentioned in [45] the modelling of the physical layer is crucial to WSN simulations. The actual modelling used will affect the performance of the higher layers. The absolute and relative performances of these simulations may differ. Another article [46] looks at the level of detail to which simulations should model the real world. Too much detail in simulations make them slow and cumbersome to implement and too little detail results in unrealistic simulations.

All of this is mentioned again in [41] which extends the research further. The authors simulate the Flooding routing protocol on three different simulators, GloMoSim, OPNET and NS-2. The advantage of using the Flooding protocol is that the implementation is very simple and well defined. They set up the networks and use the 802.11 PHY and MAC layers for their wireless communications. Some of the results obtained in this can be seen in *Fig.* 4.1, *Fig.* 4.2, *Fig.* 4.3 and *Fig.* 4.4. The widely variable results point to the exact question regarding the accuracy of simulators and their ability to portray reality.

From these figures we can see that their results vary dramatically. The results are barely comparable, and worse yet is the fact that it is unknown as to which simulator reflects reality the best. The following statements are made with regards to these differences.

- Overly simplified propagation models are used.
  - Environment is too complex to model.
  - Realistic models are too computationally intensive.
- Physical layer models differ in details.
  - Different abstractions
  - Different simplifications
- Each simulator offers different means to implement new protocols with different degrees of freedom.
- Errors in 802.11 implementations.

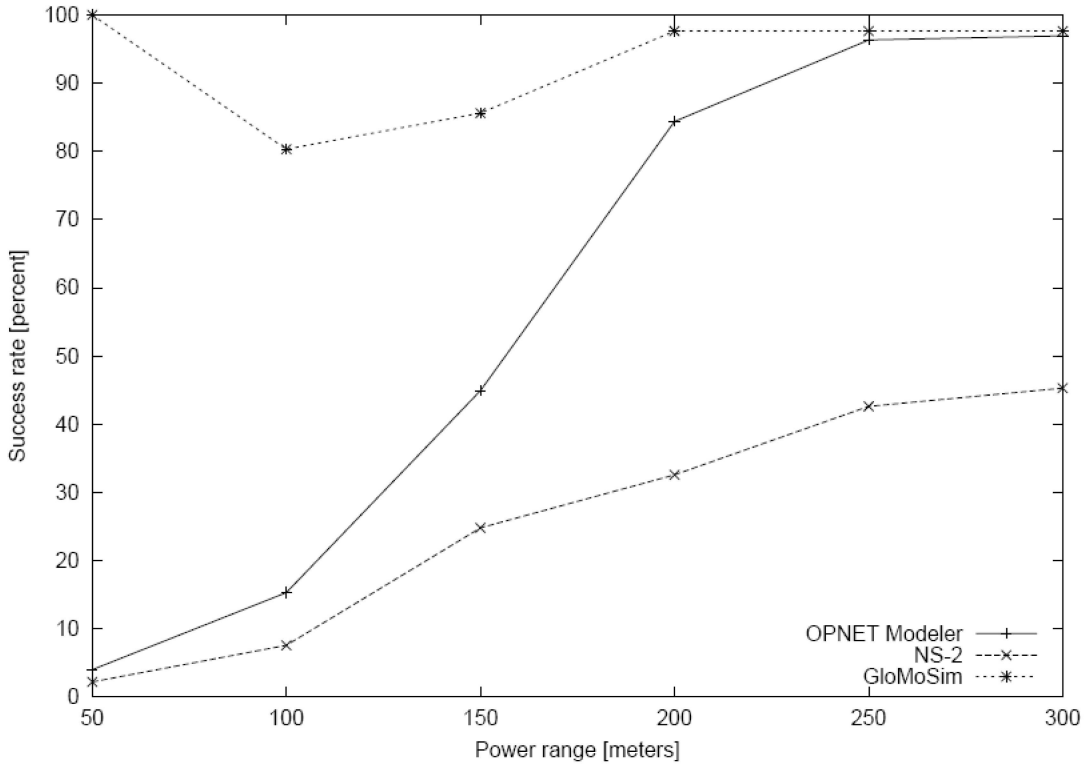

FIGURE 4.1: Success Rate vs Power Range. Taken from [41] (Fig. 3.).

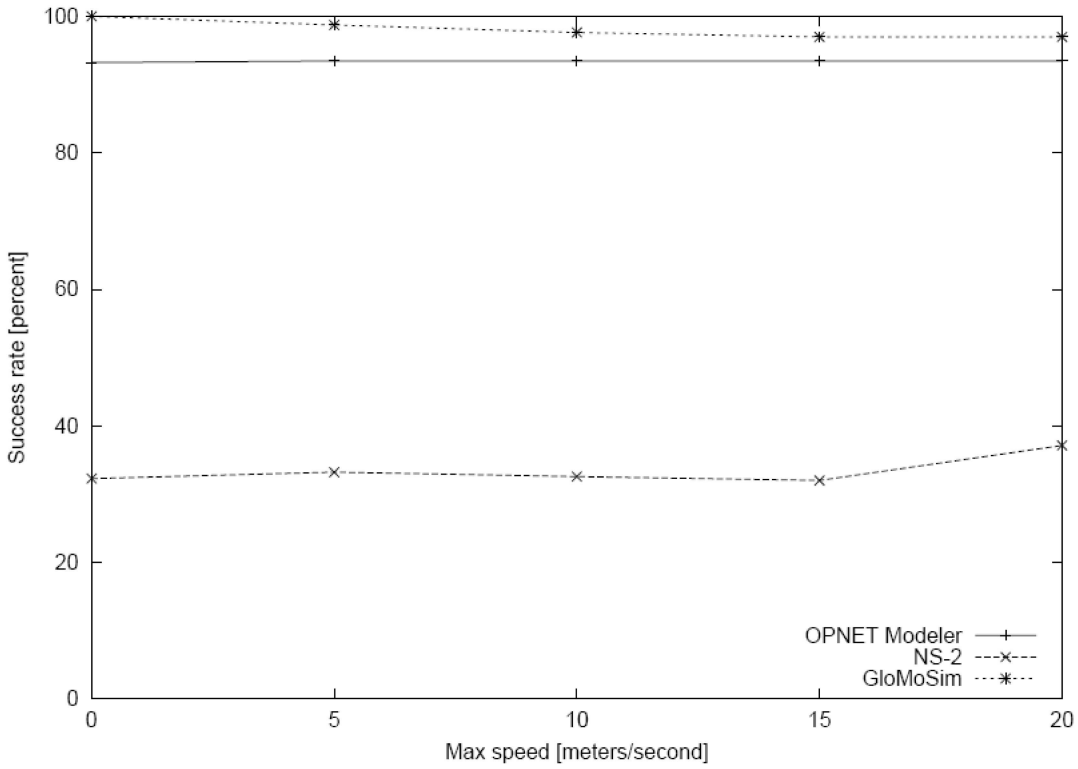

FIGURE 4.2: Success Rate vs Mobility. Taken from [41] (Fig. 4.).

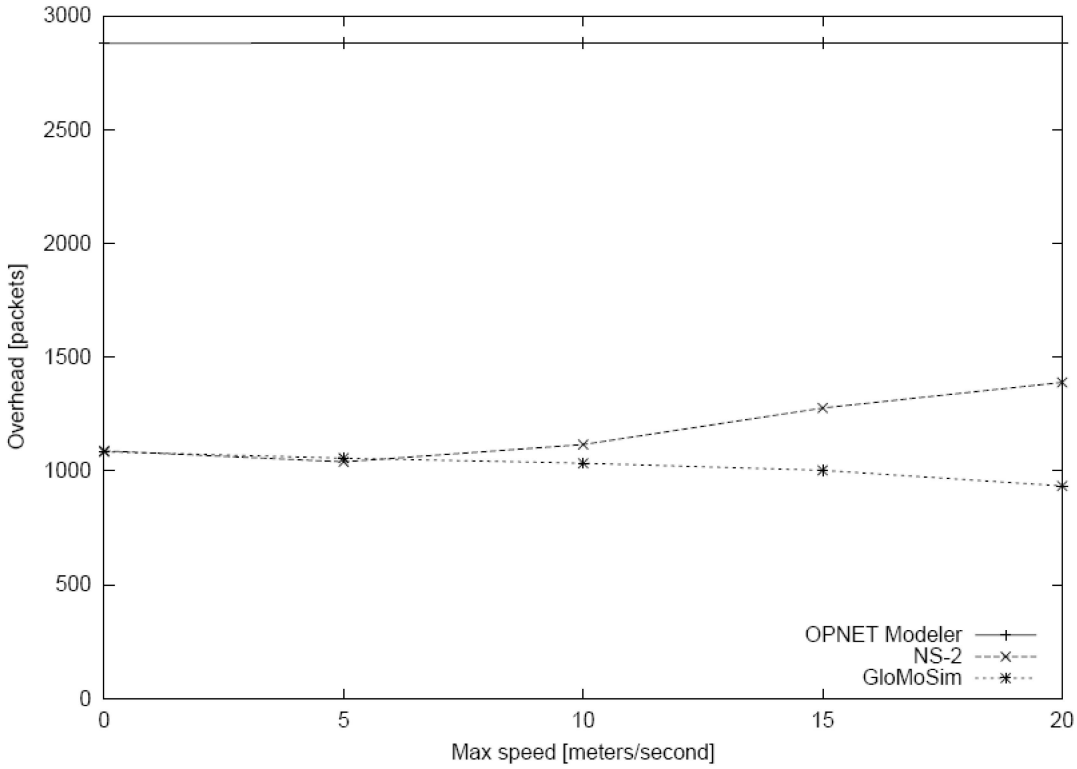

FIGURE 4.3: Overhead vs Mobility. Taken from [41] (Fig. 5.).

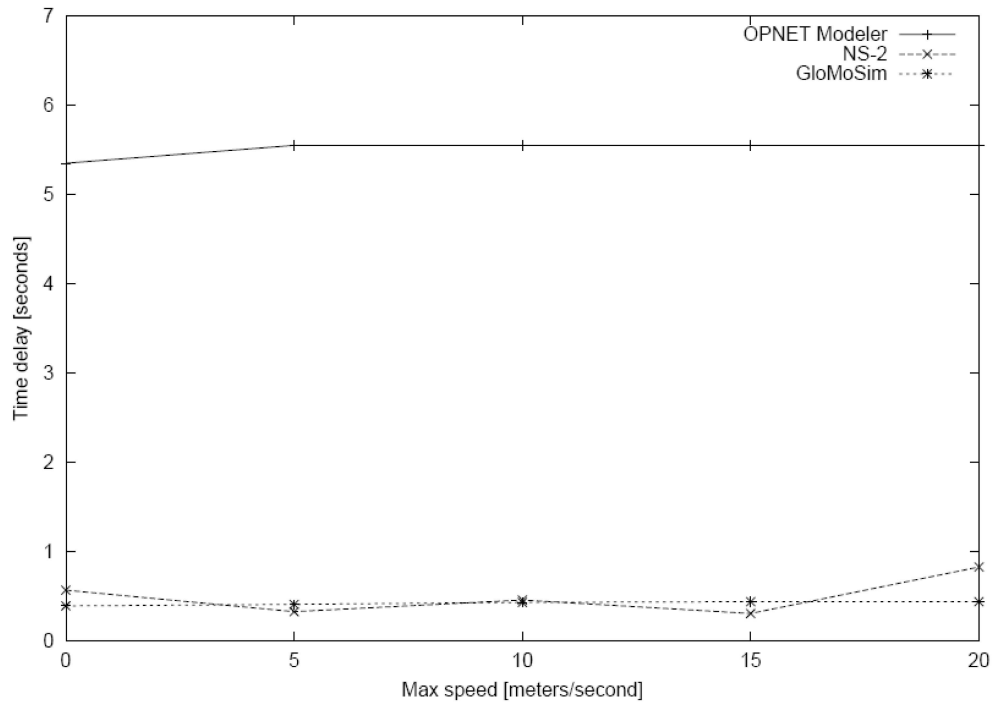

FIGURE 4.4: Time Delay vs Mobility. Taken from [41] (Fig. 6.).

The authors have suggested that a hybrid model should be used instead of pure simulations. The lower layers (PHY, MAC) and mobility should be simulated and the higher layers should be tested on physical static clusters. The article provides all the technical details that were used in their simulations, which allows for the provision of credible conclusions. One conclusion made is that not all simulators are created equal and that more research needs to be done to correct the implementations of these simulators.

4.3 PROTOCOL CREDIBILITY

The previous section focused on the simulators themselves. They have their problems, but this "lack of credibility" extends to improper simulation practices of actual protocols. In [47], the authors studied 114 peer-reviewed manet research papers published between 2000 and 2005 at one venue (SIGMobile). The results of their research survey can be seen in *Fig.* 4.5. Some of the specific examples of what they found include that:

1. 85 percent of the papers were not independently repeatable due to a lack of documentation.

2. Of the 58 papers that listed publically available simulators, 87.9 percent didn't document a version.

The results of this survey show that some form of framework or standard needs to be set up whereby authors provide the basic information that is required to make the research credible. The responsibility also lies with the conference and journal reviewers as well as the publications' readers to spot, correct, and comment on the research published.

As mentioned before, simulations make certain decisions and take some assumptions into account. These obviously affect the accuracy and precision of the results obtained. Some examples of these imprecise assumptions as listed in [48] are:

- *Transmission range is a critical factor in many manet protocols, but its characteristics are not precisely defined. Rather, investigators generally represent transmission distance as a circles radius.*

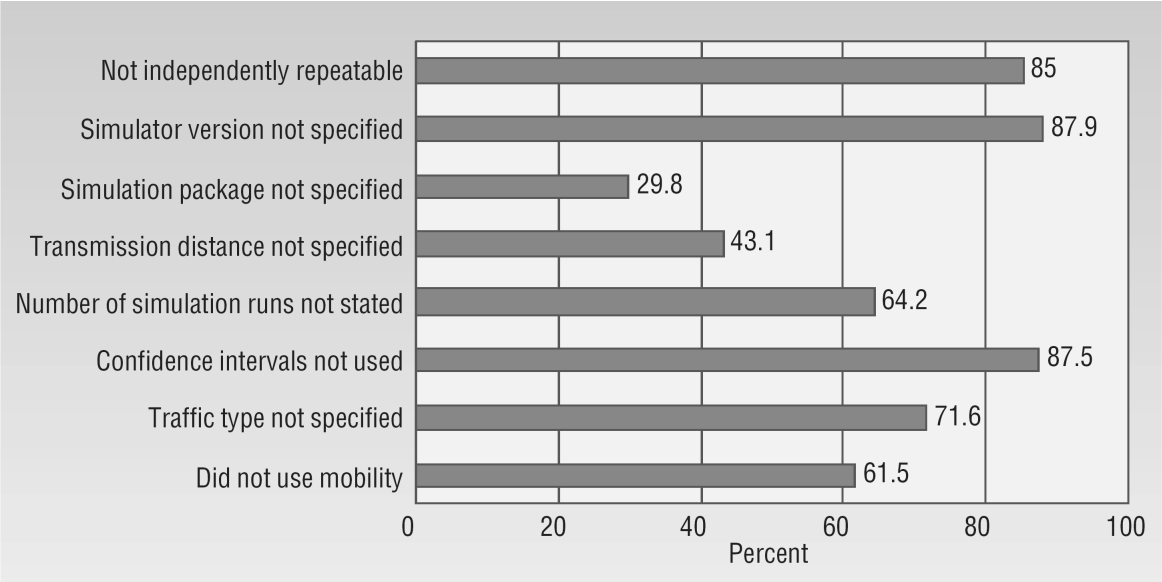

FIGURE 4.5: Manet simulation issues. Taken from [48] (Fig. 2.).

- *Researchers commonly model node distribution as uniform or random. In reality, roads, trees, water, and other obstacles affect node distribution.*
- *Interference models are typically based on SNRs or BERs. This neglects interference based on increasing traffic or unpredictable background noise.*
- *Researchers typically assume that node communication is bidirectional. However, unlike wired implementations, wireless communication doesn't guarantee signal transmission, and reception distances are equivalent. Manet nodes might have different power reserves available for transmission.*
- *Researchers commonly model node mobility as random, but it rarely is. For instance, a group of soldiers will commonly follow a preplanned path, or at least travel in the same general direction. Individuals rarely travel in random directions, pause for random times, and then embark in a completely different direction. Instead, they usually follow some pattern.*
- *Simulations typically model a square or rectangular network area. Although convenient, this rarely reflects reality. When a node reaches the network edge, does it abruptly turn or continue moving and fall out of the network area? If the latter, the node moving outside the network area would still be in transmission and interference range to nodes on the simulation areas edges.*

The research community needs to *"police our science or lose our credibility"* [48]. Some of the recommendations made by [48] are given shortly. Although not a comprehensive list, it does provide some basics guidelines to start improving the credibility of the research. Due to factors such as the *development stage, complexities and available resources* it might prove impossible to incorporate all of these recommendations.

- *Lack of independant repeatability: Properly document all settings. Publication venues have limited space, so typically include only major settings (such as transmission distance and bit rate). Provide all settings as external references to research Web pages, which should include freely available code/models and applicable data sets.*
- *Lack of statistical validity: Determine the number of required independent runs. Address sources of randomness (such as pseudorandom number generators) to ensure simulation run independence. Collect data only after deleting transient values or eliminating it by preloading routing tables and traffic queues.*
- *Use of appropriate radio models: Free-space radio models are sufficient during early model development, but two-ray and shadow models provide a more realistic environment during data collection and analysis. Tune settings against an actual implementation when available. Improve radio model abstractions as more implementations and experimental manet testbeds become available.*
- *Improper/nonexistant validation: Validate the complete simulation (developed protocol, traffic, radio model, and scenario) against a real-world implementation. When this isn't possible (such as during early concept development), validate the simulation against analytical models or protocol specifications. The latter will be less precise, but you can further refine it as implementations are realized.*
- *Unrealistic application traffic: Simple constant-bit-rate traffic might be unrealistic. Base traffic generation on intended applications.*
- *Improper precision: Use manet simulations to provide proof of concept and general performance characteristics, not to directly compare multiple protocols against one another.*

- *Lack of sensitivity analysis: Sensitivity analysis can identify a chosen factors significance (parameter settings that change in a study). For example, if youre testing two routing protocols (such as Ad-Hoc On-Demand Distance Vector and Dynamic Source Routing) against three mobility speeds, using the analysis of variance (ANOVA) technique can determine if the output changes are due to the routing protocol, the mobility setting, both, or neither. Raj Jains book [49] lists procedures for performing ANOVA calculations.*

4.3.1 SEER's Credibility

During the literature study, it was found that the validity of the results obtained by various researchers is questionable. The research conducted in this paper builds on from [5], and thus this section will focus on some of the inconsistancies found therein. It should be noted that the author received SEER's implementation from the author of [5], no mention was made as to whether the received implementation was the final version, but it was assumed as such. The design of SEER is for the most part efficient and valid, but some concerns are raised as to the validity of the testing approach taken in [5]:

- *Radio Model Distance:* The radio model that was selected is as used in [9]. The actual implementation of the protocol is suspect. Although the network that was used for simulations places the nodes one metre apart, the nodes are actually able to communicate with a diagonally opposite node. This fact adjusts the distance between the nodes from 1 metre to $\sqrt{2} \approx 1.414$ metres. This discrepency would, over time, decrease the sending nodes energy reserves quicker.

- *Network Header:* The network header is quite well defined in the research, but a field specifying the "type-of-message" that has been sent or received is missing. Although it should be noted that in SEER's current format this may not cause a problem as each of the possible messages has a unique size and can be detected in this manner. All of these additions to header adjust the overall message sizes, decreasing the energy efficiency of SEER as the sent packets are larger.

- *Power Calculations:* In the power calculations performed to determine the value by which to decrease the energy resource, the message size only consists of the network header and the actual data to be sent. The headers of the lower and upper layers are not

brought into the calculation. Although not a realistic implementation, all the power consumption calculations for all the simulated protocols are done in this manner. This at least levels the playing field to a certain degree.

- *Protocol Implementation:* With a WSN being extremely dependant on the application that is being served, it is difficult to select a routing protocol that caters for a diverse range of applications. The type of application chosen, sending periodic data messages, for the simulations places pressure on the author's implementation of all the routing protocols concerned. In the case of [5], the protocol implementations for Directed Diffusion, Flooding and SPIN were not entirely correct. Assumptions were made as to how they would react in the situation being simulated. This resulted in them performing poorly, sometimes as badly as Flooding. These protocols have been shown to perform better that Flooding in various research papers [28, 29].

- *Results:* When one considers the results shown in [5], SEER appears to out perform all the other protocols by a large margin. As mentioned previously, the results seem suspect until independently proven, as the protocols were not correctly chosen and their implementations were not correct. This paper intends to prove or disprove the results that were obtained.

4.4 CHAPTER SUMMARY

Recent concerns regarding simulator and protocol credibility were discussed. The majority of these concerns can now be applied to the simulations conducted in this research.

Inconsistancies found in some WSN research protocols were also pointed out. The fact that 114 papers all suffer from some of the same faults is indicative of the aforementioned validity. A brief look into research that was conducted for SEER found some errors and assumptions. These were investigated and the corrections have been made for the simulation implementations as can be found in chapter 6.

Having looked into these concerns, it is now possible to design a new protocol and incorporate the implementation and testing guidelines mentioned previously.

CHAPTER FIVE
PROTOCOL DESIGN

5.1 CHAPTER OVERVIEW

This chapter describes the design of a routing protocol that incorporates features found in both flat and hierarchical protocols. The design and implementation of this *Hybrid Energy Efficient Routing* protocol is discussed.

5.2 PROTOCOL DESIGN CHOICES

As has been discussed in previous chapters, wireless sensor network protocols should exhibit a few necessary design features. How many of these features are incorporated into a design will affect various aspects, but namely energy efficiency. The goal of the design for HEER was to incorporate those features that will allow for an efficient routing protocol, but could also adapt to the requirements of the WSN environment. Listed below are the features that were selected for inclusion in the routing protocol.

5.2.1 Energy Efficiency

Definitely the most important design criteria that should be taken into account with any component being designed for a WSN environment. As mentioned in [10], communications in a wireless sensor node is the single greatest user of the power available to a node, up to 75%. This fact dictates that all communications in a WSN should be minimized. With this in mind, HEER was designed to incorporate the following features in an effort to reduce power consumption and to maximize network lifetime.

1. *Source initiated / Event-Driven:* Should a node have information to communicate to the sink node, the node will initiate communication. The sink node(s) do not

have access to address the nodes directly. This feature minimizes the number of transmissions that are distributed into the network.

2. *Routes are dynamically established:* The pathway from one node to another is calculated on the fly.

3. *Computationally simple:* The route calculation consists of a limited number of comparisons based on information that has been built up during the operation of the protocol. No complex calculations are required.

4. *Data aggregation:* An original feature of a hierarchical protocol, where a cluster head accumulates the information sent by its member node, information from nodes is aggregated the closer it comes to the sink node.

5.2.2 Reliability

In most applicable applications there exist two types of messages, critical and non-critical. For critical messages to be effective and timeously acted upon they need to arrive reliably. This requirement puts more pressure on a protocol to effectively manage this. The use of acknowledge (ACK) messages for all communications would not be recommended as this would introduce more transmissions on the network. The use of ACK messages for only the critical messages would be a better solution. SEER sends critical messages to two nodes instead of one. This increases the possiblity of the message reaching the sink, but does not guarantee it. By using ACK messages, which are smaller than data messages, we can introduce reliability into the design.

5.2.3 Scalability

Wireless sensor networks will one day be all around us, similar to smartdust [2]. For this prediction to become a reality, new nodes need to be dynamically included into the network without any input from an outside source. The network needs to operate efficiently and routing protocols should take this into account. HEER was designed with this in mind. For maximum scalability a flat architecture was selected for the foundation. Hierarchical networks usually require specific clusterheads to be distributed as well as normal nodes. This limits the scalability that can be achieved.

5.2.4 Numerous Sink Nodes

As the network size increases, the nodes surrounding the sink are placed under greater amounts of stress as they handle more messages. To improve the performance of extremely large networks, the ratio of sink nodes to normal nodes should be increased. The routing protocol needs to be able to handle this particular situation.

5.2.5 Hardware independence

A protocol should be designed for independence from any one particular node technology or requirement. This facilitates the ability to operate on various platforms. Protocols such as LEACH are designed with the assumption that there is a high power transmitter that can reach the node or that the node is within transmission range if set to full power. This dual redundancy increases the cost of construction of the nodes.

5.3 PROTOCOL OPERATION

The following section describes the actual design of HEER. It should be noted that all nodes require memory for a neighbour table (containing neighbour node information), and a branch-table (containing branch-tree members). The different phases or stages of HEER are covered next. All node addressing in the protocol uses a 16-bit addressing scheme, which allows for 2^{16} (65536) unique addresses. All times used in HEER (4 hour broadcast messages and 5 minute Join or Info messages) could be extended, but for the limited energy simulations conducted, these times resulted in the best performance.

STEP 1: Network Discovery and Setup

Similarly to SEER [5], once the network has been established in its operating environment the sink node transmits a Broadcast message, whose header is shown in *Table* 5.1. The sink node chooses a sequence number that allows the nodes to distinguish between new and old broadcasts. The hop count is initially set to zero with each node adding to this value when forwarding the broadcast. A node will also insert its current energy level and hop count into the message. The broadcast message serves the main function of allowing the nodes to generate a neighbour table from their surrounding nodes. A node will broadcast any broadcast message only once, but can receive up to n broadcasts, where n is the number

of neighbours connected to the node. A sink node will transmit this broadcast message every four hours.

Field	**Size (bits)**
Source Address	16
Destination Address	16
Energy Level	16
Hop Count	8
Sequence Number	8
kind	4
Total	68

TABLE 5.1: Fields contained in the network header of a Broadcast Message

For each entry into the neighbour table, the node calculates an administrative distance value for the corresponding neighbour node. This "admin" value is calculated as follows:

$$admin = \frac{(H_c)^{-d}}{P_n} \tag{5.1}$$

where H_c is the hop count of the neighbour, d the distance calculated from the received power (P_r) using *Eq.* (6.1), and P_n is the current energy level of the neighbour node. The distance calculated is more a "psuedo" distance, as this value is affected by environmental issues. HEER uses a form of tree structure for its routing scheme. The node will select a "branch"-head, the node with the highest admin value, from its neighbour table. After five minutes from receiving the initial broadcast, the node will send a message to its' branch head informing him of the joining. This is done using a Join message, whose header is shown in *Table* 5.2. No information is sent with this message, but the destination address is used by the branch head to construct a branch-table that has all the branch-nodes that currently "grow" from the branch-head.

After another five minutes, any node that has become a branch-head, i.e. has nodes branching from it, will send out a broadcast message structured as the Info message format shown in *Table* 5.3. This message has the number of branches that currently belong to the branch-head.

Field	Size (bits)
Source Address	16
Destination Address	16
Kind	4
Total	36

TABLE 5.2: Fields contained in the network header for the Join Message

This information is used to reroute messages at a later stage.

Field	Size (bits)
Source Address	16
Destination Address	16
Branch Size	4
Kind	4
Total	40

TABLE 5.3: Fields contained in the network headers of the Info message

STEP 2: Transmitting Data

Once the set up of the network has taken place, the transmission of normal data messages can occur. Like SEER, HEER also allows for critical messages, e.g. sharp rise in temperature. Data messages are sent every 15 minutes, with critical messages occuring after every 10 data messages. Data messages are structured as in *Table* 5.4. The Creator ID field is used to keep track of the originator of the sensed event. Data messages are also used to update neighbour tables, hence source, destination, energy level, and hop count variables are adjusted accordingly. The critical field represents whether the message is critical or not. If so, appropriate action can be taken (discussed later). The Message Size field is used for limited aggregation. As messages travel towards the sink, the messages are aggregated, with messages coming from the edge of the network first. As a message arrives at a branch-head, the header is stripped (except for the data and Creator ID fields) and the current node's header and data is added. This field allows a receiving node to know how big a message is for reception and transmission.

Field	Size (bits)
Data	32
Creator ID	16
Destination Address	16
Source Address	16
Energy Level	16
Hop Count	8
Message Size	8
Kind	4
Critical	1
Total	109

TABLE 5.4: Fields contained in the network header of a Data Message

Field	Size (bits)
Source Address	16
Destination Address	16
Kind	4
Total	36

Table 5.5: Fields contained in the network headers for the Acknowledge (ACK) Messages

Critical information is handled in a different manner. A critical message is sent immediately to the branch-head. Which will forward the message to its branch-head and so forth until the message reaches the sink node. Small acknowledgement messages, as shown in *Table* 5.5, are sent in return upon the receipt of a critical message. Should the sender of the critical message not receive an ACK message in a certain time frame, the message will be resent. Should it fail again the message will be broadcast. Only nodes that have an equal or smaller hop to the sink will respond. Although the sending of an ACK does increase the number of messages that need to be sent, the ACK adds the reliability functionality of the protocol.

STEP 3: Forwarding Data

As mentioned previously, a data message is sent to a branch-head. This branch-head will then compile the message together with its own message and the other messages from its other branches. This message will then be sent to the next branch-head. Each time the sent message contains the energy and hop count information of the corresponding neighbour node. This information is used to calculate a new admin distance as well as to update the neighbour table.

STEP 4: Network Maintenance

To maintain, facilitate and pre-empt a routing failure, HEER makes use of a number of features. Firstly, a Power message, *Table* 5.6, is sent when the node is about the fail. This message is broadcast to all its neighbours informing them of its imminant demise. If the node belongs to a specific branch-head, this head will remove the node from its branch-table. If the node is a branch-head for other nodes, they will seek out alternative routes. This brings us back to the Info Message, seen after every broadcast message sent by the sink. As mentioned this Info message is used to inform all neighbours about the number of branches present at that node.

If a node has lost its branch-head, it will search through its neighbour table searching for another node to attach to. To increase the overall network lifetime, a node will join the node with the lowest number of branches. This will ease the pressure on nodes with a large number of branches.

Field	Size (bits)
Source Address	16
Destination Address	16
Kind	4
Total	36

TABLE 5.6: Fields contained in the network headers for the Power Messages

5.4 CHAPTER SUMMARY

The design choices that were chosen and their implementation in HEER were discussed. The designed protocol incorporates features of both flat and hierarchical features. The results and discussion regarding HEER's performance are covered in chapter 7.

The HEER protocol can be summarized as follows:

1. The sink node initialises the network using a Broadcast message.
2. Nodes add all neighbours to their neighbour table. Nodes will calculate an "admin" value for each neighbour and will select the node with the highest value.
3. Nodes will then inform their branch-heads that they are joining them, this is done using a Join message.
4. Branch-heads will then inform neighbours about the number of branches they currently have.
5. Node data is then sent to a branch-head where it is aggregated with the branch-head's message as well as with the other branches (if present).
6. Critical messages are sent to the branch-head, but are not aggregated. They are then sent from one head to another until the message arrives at the sink. Acknowledge (ACK) messages are used to ensure that the message arrives reliably.
7. If a node's energy is very low, i.e. node is about to die, the node will send a Power message informing neighbours accordingly. Neighbours will then adjust their routes by either removing the node from their branch-table or selecting another branch-head.

8. Sink node will send periodic broadcast messages to maintain accurate hop count values and allow for network restructuring due to inactive nodes.

CHAPTER SIX
PROTOCOL EVALUATION

6.1 CHAPTER OVERVIEW

This chapter describes the evaluation procedures that were undertaken during the implementation and design of the simulation environment. The various types of methods in which the research can be conducted are discussed, followed by the actual implementation as well as the criteria for the results used.

6.2 VERIFICATION

Three methods exist which allow us to test a designed systems' performance; analytical, simulation or physical implementations. These are further discussed below.

6.2.1 Analytical Methods

An analytical method requires mathematical formulae to determine the performance of a system. A method of this nature can be simple or complex depending on the level of assumptions. If no simplifications or assumptions are made, the calculations can become extremely complex. If taken to the extreme, one could argue that a single beat of a butterfly's wings could adjust the performance of the system, a common practice in chaos theory. The advent of personal computers and their ability to do complex computations has simplified this process somewhat. Eventually a decision has to be made and a line drawn about the assumptions and real world criteria that will be applied to the analysis.

6.2.2 Physical Implementations

Implementations on physical nodes can, and do prove, that a protocol could operate on real world equipment. The protocol is programmed in the appropriate language and loaded onto

the node. Once this is has been accomplished on all the nodes, the selected experiments can be conducted. This particular method is best for examining the protocols' interaction under real world conditions. The wireless channel is "modelled perfectly" and power consumption would be more realistic. The problem with this method is the fact that physical equipment is required. Wireless sensor nodes are still expensive when large quantities are considered. If the idea of *Smartdust* is to be realised, the cost of these nodes will need to be decreased.

6.2.3 Simulator Implementations

Due to the costs involved in a physical implementation, simulations have become the foremost technique to test WSN theories. Simulators operate on PCs and in effect use some of the calculations that are formulated and used in an analytical analysis. Simulators suffer from the same assumption and calculation errors that may be present in analytical methods. Another aspect to consider is the credibility of the chosen simulator. The assumptions made can have a dramatic effect on the outcome of the results.

6.3 SIMULATION IMPLEMENTATION

The following sections describe the setup for the simulator.

6.3.1 Simulator Setup

Omnet++ [42, 43] (version 3.2) was chosen as the underlying simulator with the Mobility Framework [44] (version 1.0-a-5) forming the core wireless environment. The chosen protocols, discussed later, were implemented in C++ (Visual Studio .Net 2003). The number of simulation runs were increased to allow for greater statistical validity with the seeds for the PRNGs being initialized to a new variable each time.

6.3.1.1 Assumptions and Decisions

Physical Layer: Besides the requirement of wireless communication placed on the physical layer there are a few others. Some of these are listed below.

- The transmitter power needs to be variable.

- The RSSI value of the received message needs to be made available to higher layers for their routing calculations, in this case HEER.

Medium Access Control Layer: Certain assumptions have been made regarding this layer. The MAC protocol needs to support the following functions:

- Support for both CSMA and TDMA operation: Flooding and SEER require CSMA for all their communications, but HEER and LEACH require TDMA functionality after the initial setup phases. It is further assumed that this layer will be responsible for transmitting the required time slice information to the respective nodes.
- Broadcasting of a message needs to be possible. Not all protocols at this layer support this function. Since all the protocols broadcast a message at some point, this specification is a must.
- The radio is available when needed, i.e. not switched off or in a sleep mode. Although this may not be a realistic assumption, this research is mainly aimed at the network layer.

Application Layer: A simplistic application layer was defined for the simulations. A data message is generated every fifteen minutes and passed onto the network layer. After ten data messages have been sent a critical message is then created and sent. This process repeats itself until the node has depleted its energy source.

Channel Model: A free-space-loss model is used to model the channel [50]. This model is explained as follows:

$$P_r = \frac{P_t}{(\frac{4\pi}{\lambda})^2 * d^2} \tag{6.1}$$

where P_r is the received power, P_t the transmitted power, λ is the wavelength of the signal and d is the distance between sender and receiver. This model is predominately used in line-of-sight (LOS) systems, which is not always the case with a WSN. To increase the validity of the model, *Eq.* (6.1) is modified to *Eq.* (6.2).

$$P_r = \frac{P_t}{(\frac{4\pi}{\lambda})^2 * d^{\alpha}} \tag{6.2}$$

where α is referred to as the path loss exponent (PLE). This variable better describes reality by taking various signal environments into account and by adjusting the sensitivity of the received power to the distanced traveled. It has been shown that values of $2.3 - 2.6$ are valid for typical outdoor conditions [51]. Since this is the more appropriate operating area for a large WSN, an α value of 2.5 was chosen.

Radio Model: Arguably the most important part of the simulator, the radio model selected defines the extent to which the transmission and reception of messages will deplete the energy source. A first order radio model was selected for these simulations. This model was initially used in [9] to calculate the power consumption of LEACH. The radio model was subsequently implemented in [5]. The energy for transmission (E_{TX}) is calculated by (6.3):

$$E_{TX} = E_{elec} * k + \epsilon_{amp} * k * d^2 \tag{6.3}$$

and reception (E_{RX}) by (6.4):

$$E_{RX} = E_{elec} * k \tag{6.4}$$

where E_{elec} is the energy consumed by the transciever electronics, k is the bit size of the data message, ϵ_{amp} is the energy consumed by the transmitter amplifier and d is the transmission distance in metres. As in [5, 9], E_{elec} was set at $50nJ/bit$ and ϵ_{amp} was set at $100pJ/bit/m^2$. The same transmission distance was used for all the simulated protocols, except HEER where the transmitter is able to adjust its transmisstion power. Each of the nodes had their energy sources initialized to 5 mJ. The reason for this was to limit the overall simulation time and reduce the computational resources required. An assumption made regarding the radio model is that the radio channel is assumed to be symmetric. This means that the same amount of energy is required to send a message from one node to another as it is to send a message back. Although this is not valid in a real world scenario, this assumption was made to cater for the limited resources available in the current simulators. For all the simulations a radio

carrier frequency of 868 MHz and a signal attenuation threshold of −110 dBm was selected. These values were based on the CC1100 chipset available from Chipcon [52].

The radio model is not perfect as it makes a few assumptions. It assumes that reception is only related to the distance between two nodes and does not take into account channel irregularities.

Mobility: The simulator is set up where most of the network is stationary. A pre-selected amount of nodes are created as mobile nodes. These nodes slowly move randomly around the network. Random walks are not really true to simulations, as soldiers would most likely follow a pre-determined path for example (as mentioned in chapter 4). Due to the nature of the routing protocols selected, and HEER being the only protocol that supports any form of mobile nodes, it is the only one simulated with mobiles nodes. This is done to compare network results with mobile nodes to a network with only stationary nodes. The number of mobile nodes were limited so as to not impact the comparision to severely.

Node Hardware: As mentioned above, a variable transmitter is a requirement for HEER. One more requirement would be the need for storage memory, preferably having a fast access capability, i.e. RAM. The memory usage for HEER is definitely greater than SEER, but not excessively so.

Cross-Layer Communication: For HEER to function effectively, the different protocol layers need to be able to communicate with each other. HEER, sitting on layer 3, requires information from, and control over, lower layers. Interaction between HEER and the application layer is also crucial for optimum performance.

6.3.1.2 Credibility

The concerns raised in chapter 4 have been taken into account when setting up the simulator. With simulator credibility and the current accuracy of WSN research questioned, it is important to ensure that accurate and repeatable results are achieved.

6.3.2 Network Setup

Two network layouts were chosen to simulate the selected routing protocols on, these are discussed below. The network sizes chosen included: 25, 50, 100, 500, 1000, 1500, 2000, 5000 nodes.

Uniform Network: Designed with the sink node in the middle of the network, the uniform network places the nodes in a square formation. Each node is equi-distant from each other, with the distance between them set at one metre. Each node can have up to eight neighbours arranged around them. The nodes in a uniform network can be seen as creating ever expanding circles away from the sink node, like a two dimensional slice of an onion. These circles or layers would thus each be one more hop away from the sink node. The uniform network assumes the worst case routing scenario where each node is only connected to the next layer to and from the sink. This means that the all messages must go through the next layer that is closer to the sink with no possibilities for jumping a layer. With all the nodes effectively being the same distance from each other, HEER will be the most affected as its routing decisions are based partly on the distance between nodes. An example of the uniform network is shown in *Fig.* 6.1.

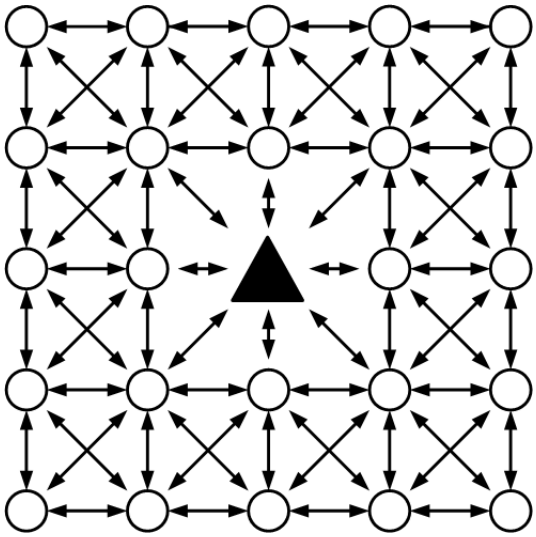

FIGURE 6.1: Example uniform network layout showing connectivity.

Random Network: To give a better comparison of HEER's performance, a random network layout was also implemented. This network layout places the nodes in random positions on

the available network space. The mobility framework calculates and creates the connections dynamically. A example random network is shown in *Fig.* 6.2.

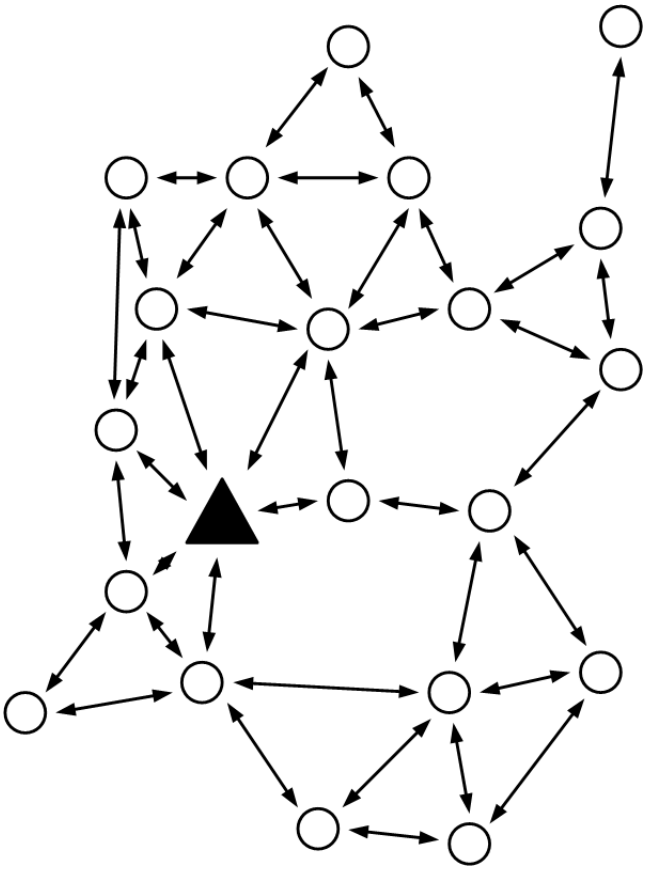

FIGURE 6.2: Example of a random network layout and possible connectivity.

6.3.3 Type of Evaluations

The tests that were selected to be used as benchmarks for the protocols are listed and presented here.

Test 1: *Time until the first node dies.* As messages are transmitted amongest the network, nodes decrease their available energy levels. The test shows the overall efficiency of the protocols' ability to function at maximum capacity.

Test 2: *Time until sink neighbour is unreachable.* All of the simulated protocols, except LEACH, rely on the nodes directly connected to the sink node. Should all these nodes expire, the sink becomes unreachable. LEACH on the other hand uses its high power tranmitter to

transmit the clusters' messages to the sink. This ability bypasses the need for the sinks' neighbours to carry the weight of the network traffic.

Test 3: *Time instant when the number of alive nodes reaches a certain percentage.* A test of this nature allows us to examine the lifetime of the network and how long it takes for the network to reach certain levels of functionality.

Test 4: *The average remaining energy of the nodes at particular times.* This test shows the energy that has been consumed on the nodes at certain time intervals, allowing us to see the energy effectivity of the protocol during operation.

Test 5: *The average number of messages that have been sent in the network at selected intervals.* A unique test that shows the number of messages that have been transmitted and handled in the network. A protocol like Flooding generates more message traffic than say SEER and this is apparent from the results generated.

Test 6: *The number of data messages received by the sink at selected intervals.* The results of this test show how many of the data messages sent by the nodes are received by the sink. For a network size of 25 nodes (1 sink node and 24 normal nodes) during each data message interval,the sink node should receive 24 unique data messages. It will also allow us to compare the number of received messages by the sink before the sink becomes unreachable.

6.4 ROUTING PROTOCOL COMPARISON

Three protocols were chosen to be simulated against HEER, these are listed below. Any assumptions and changes that were made to their implementations are discussed.

6.4.1 Flooding

The implementation for Flooding is discussed here.

6.4.1.1 Protocol Changes

A few changes were made to the Flooding protocol design:

1. A TTL field is used to limit the number hops a message can travel. The value was set to the maximum number of hops from one end to the other.

2. The initial broadcast message the sink transmits uses a sequence number field which allows the nodes to determine whether the message is a new broadcast message, and whether they should process the message.

3. All data messages are treated equally, be they critical or not.

6.4.1.2 Flow Diagrams

Simple diagrams of Flooding's operation are shown in *Fig.* 6.3 for a received message and *Fig.* 6.4 for sending a message.

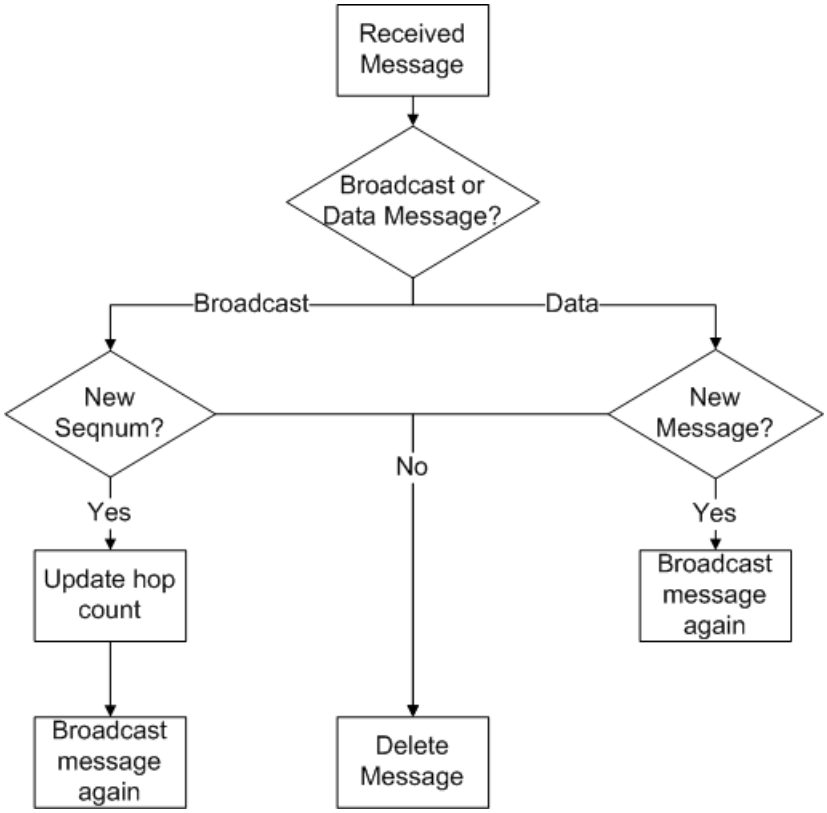

FIGURE 6.3: Receiving portion of the Flooding protocol.

6.4.2 Simple Energy-Efficient Routing

SEER was implemented as stated in [5]. The message sizes were adjusted to include a "type of message" field. This deficiency was addressed in chapter 4

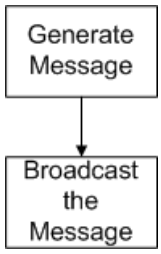

FIGURE 6.4: The Flooding protocol sending routine.

6.4.2.1 Flow Diagrams

Flow diagrams of SEER's receiving and sending implementations can be seen in *Fig.* 6.5 and *Fig.* 6.6 respectively.

6.4.3 Low Energy Adaptive Clustering Hierarchy

6.4.3.1 Protocol Changes

The original implementation of LEACH, as stated in [9], has the immediate neighbours joining the closest clusterhead. The nodes simulated do not have the ability to communicate with every other node. To this end a small adjustment was made to allow the advertisement phase to expand from the clusterhead covering nodes a few predetermined hops away, similiar to a LEACH-C implementation. This was done using a TTL field in the network header. The joining messages would then be forwarded to the next node closest to the clusterhead. This process would repeat until the message was received by the clusterhead. The clusterheads themselves maintain a list of the nodes that are members and wait for all the data messages to be received before transmitting using a high power transmitter to the sink node. Clusterhead election takes place every four hours, this facilitates the low energy that the nodes are initialised with. Critical messages are not catered for in this implementation of LEACH.

6.4.3.2 Flow Diagrams

Flow diagrams of LEACH's receiving and sending implementations can be seen in *Fig.* 6.7 and *Fig.* 6.8 respectively.

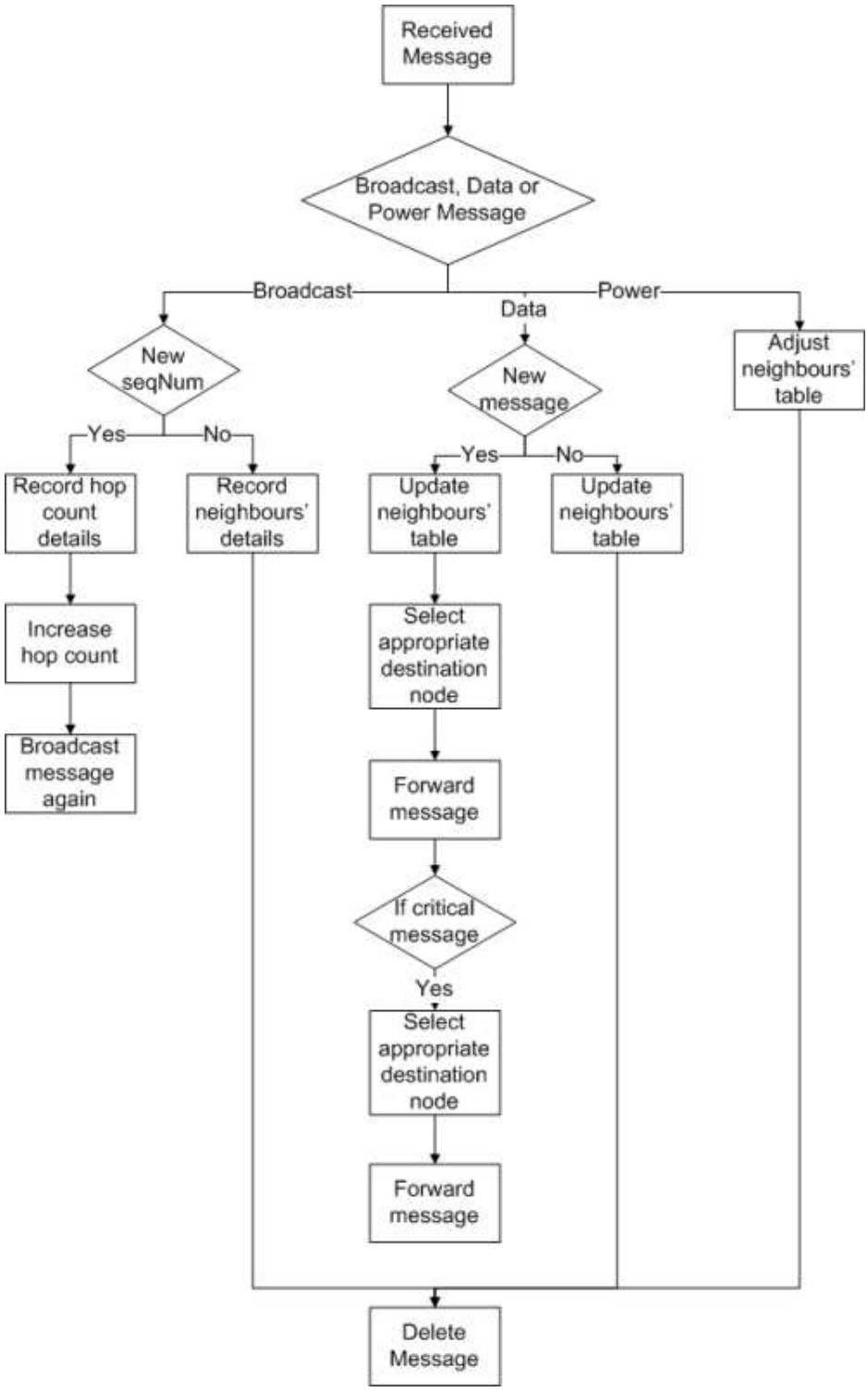

FIGURE 6.5: Receiving implementation of the SEER protocol.

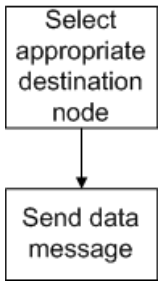

FIGURE 6.6: Sending implementation for the SEER protocol.

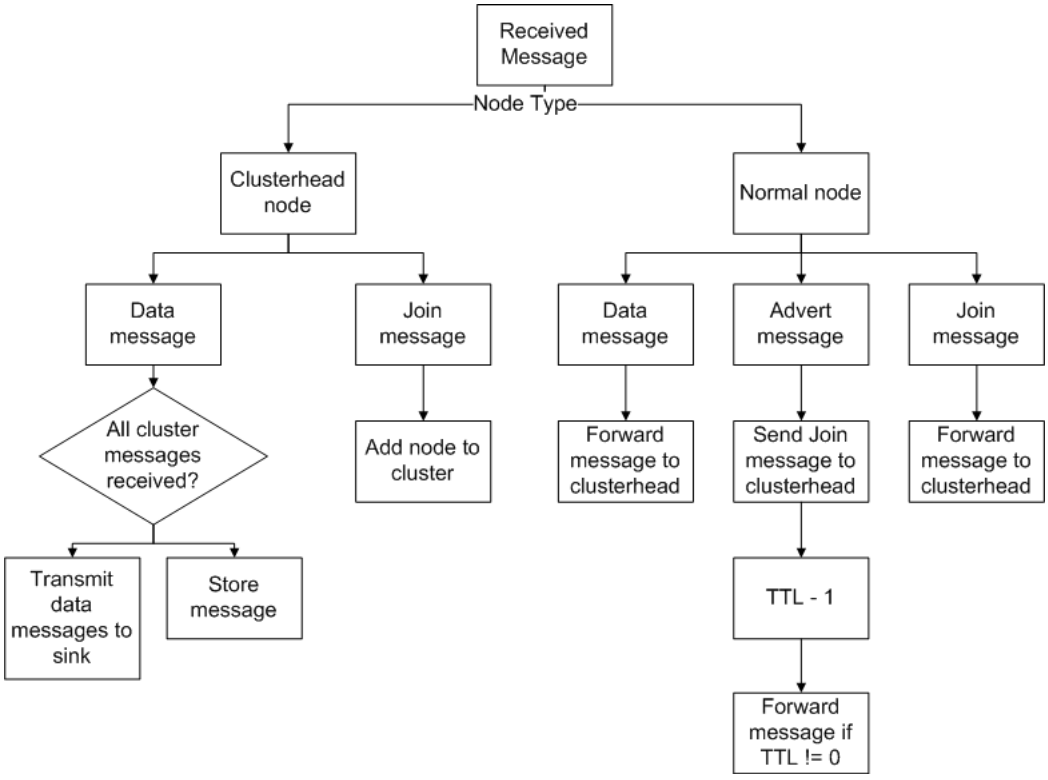

FIGURE 6.7: Receiving implementation for LEACH protocol.

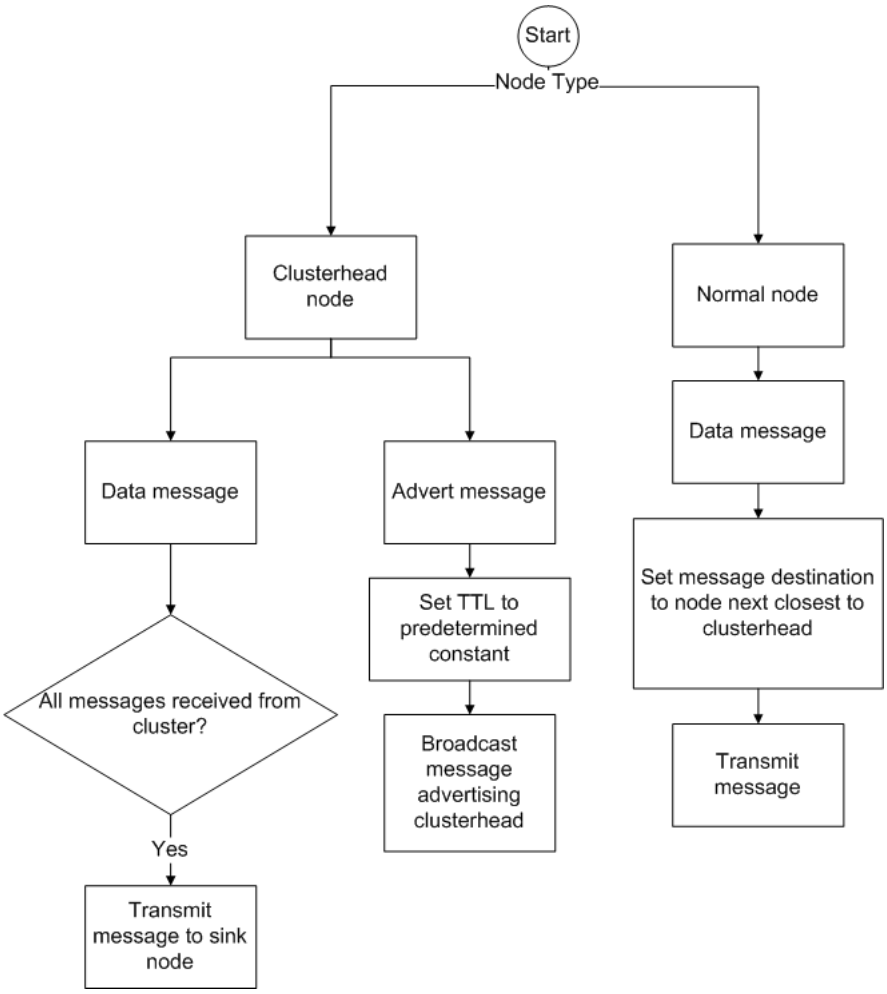

FIGURE 6.8: LEACH protocol's sending implementation.

6.5 MESSAGE HEADERS

Each of the simulated protocols make use of different network headers for the different messages, each differing by the number of bits. Table 6.1 displays the different message header sizes used in the simulations. Each data message includes a 32-bit data value.

Protocol	**Broacast**	**Data**	**Power**	**Advert/ACK**	**Join**	**Info**
Flooding	52	108	-	-	-	-
SEER	68	125	68	-	-	-
HEER	68	109	36	36	36	56
LEACH	-	100	-	68	60	-

TABLE 6.1: Network header size in bits for the various protocols simulated.

6.6 CHAPTER SUMMARY

In this chapter we discussed the simulation setup, and the decisions that were taken that resulted in this particular method of performance testing being selected. Furthermore, as mentioned in chapter 4, the problems that have been found with simulations have been addressed and taken into account in this chapter.

The routing protocols selected for the simulation were also discussed and changes to their implemenations were listed and expanded upon.

CHAPTER SEVEN
RESULTS AND DISCUSSION

7.1 CHAPTER OVERVIEW

Following the implementation of the designed protocol onto the simulation environment, the afore mentioned tests were simulated. The results of these tests and a brief discussion covering them can be found in this chapter.

7.2 SIMULATION RESULTS - UNIFORM NETWORK

The simulation environment was set up using the specifications mentioned in Chapter 6. The following results were obtained using the uniform network. The uniform network evens out the odds, as HEER is not really able to used its adjustable power capabilities.

7.2.1 Test 1 - *Time until the first node dies*:

This test shows the time that the first node in the network expired. The test was run for various network sizes. It is important to see how scalable the protocols are from this graph *Fig.* 7.1. The first node to die in this simulation is always a sink node neighbour, mainly due to the number of messages that a sink neighbour has to forward to the sink. LEACH overcomes this by direct transmission to the sink node. It can also be seen that Flooding is the worst performing protocol, as is expected. Flooding definitely suffers from excessive *implosion* in this case. Setting up the clusters for LEACH involves a number of messages being sent. Once this is complete the network is more efficient, with the clusterheads effectively forming multiple “miniature” sink nodes. SEER’s performance is comparable with the previous results obtained in [5].

When looking at HEER’s performance, it can be seen that there is a marginal gain over

SEER. This is due to better route selection and nominal gains via varying the power transmission. HEER is shown as being the most scalable of the protocols tested.

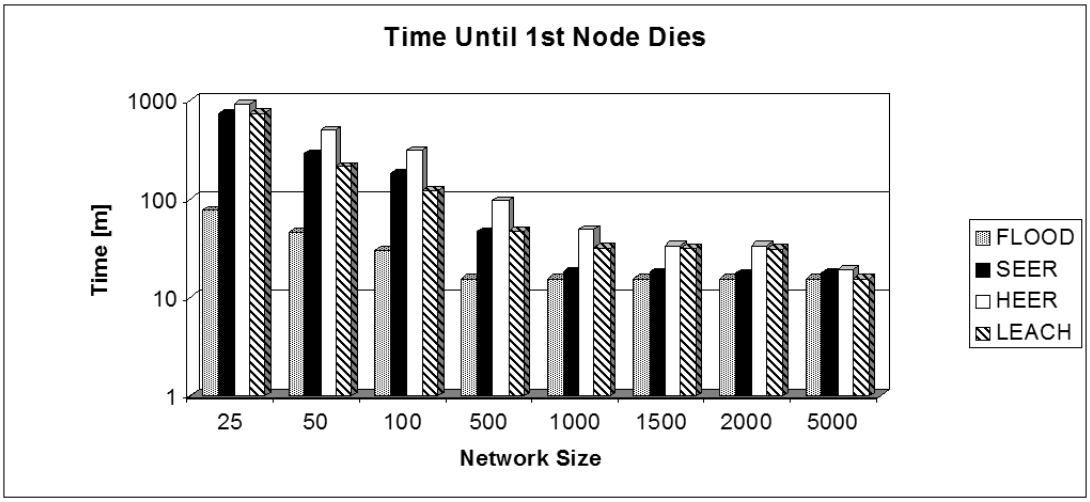

FIGURE 7.1: Time at which the first node fails due to depleting its energy source.

7.2.2 Test 2 - *Time until sink neighbour is unreachable*:

Testing to see when the sink node becomes unreachable due to all its neighbours expiring shows the excessive number of message that they forward. LEACH is not tested as its nodes have a direct link to the sink node, negating this test.

The number of messages and they way the protocols utilise them is a determining factor on sink reachability. The larger the network, the greater the number of messages that is present in the network. In *Fig.* 7.2, we can see that any network larger than 25 nodes overwhelms the Flooding protocol. It can also be seen that HEER is performing better than SEER, even with the added benefit of reliablility. The main factor in these gains is the use of data aggregation as the messages travel towards the sink node.

7.2.3 Test 3 - *Time when the num. of alive nodes reaches a percentage*:

To test the overall network lifetime, a simulation on a network size of 50 nodes was done, *Fig.* 7.3. The test involves determining the time when a certain percentage of the network is still operational. Flooding is once again the weakest performer, and HEER lasts the longest. It should be noted though that the LEACH network will allow the greatest number of messages

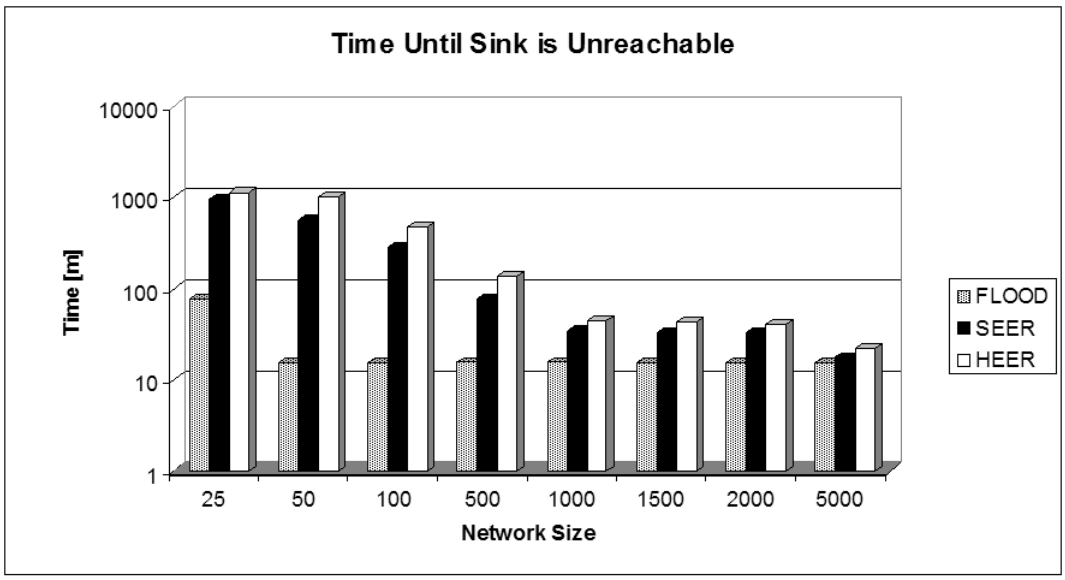

FIGURE 7.2: Time at which the sink node becomes unreachable.

to reach the sink node, due to the direct communication link with the sink node as well as the advantage of rotating the clusterheads. Once the 75% mark is reached, the sink is no longer reachable via any of the flat protocols, Flooding or SEER, or even the hybrid HEER.

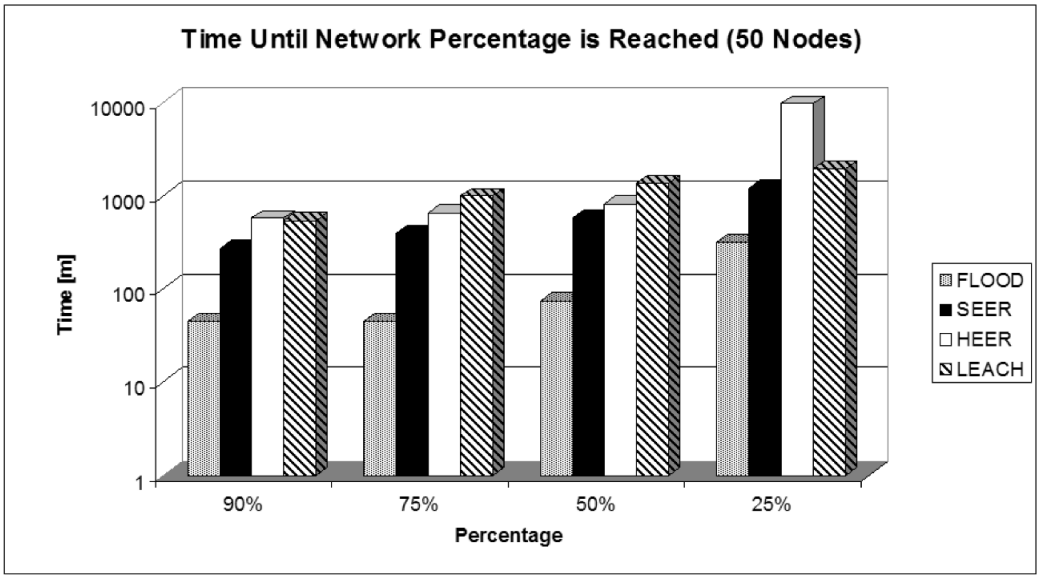

FIGURE 7.3: Time instant when the number of active nodes reaches a certain percentage in a 50 node network.

7.2.4 Test 4 - *The average remaining energy of the nodes*:

The energy that is available to the nodes is the crucial element in determining when they will expire. How a protocol manages to conserve the available resources is an effective measure of its performance. In *Fig.* 7.4, we can see how the various protocols perform. LEACH performs very well here, as the clusterheads provide a shorter path to the sink node. This quality is discussed in section 3.2.3.1. Flooding naturally proves once again that it is a poor routing protocol especially when compared to more modern protocols. HEER outperforms SEER, once again due to data aggregation, variable power transmission and more efficient routing.

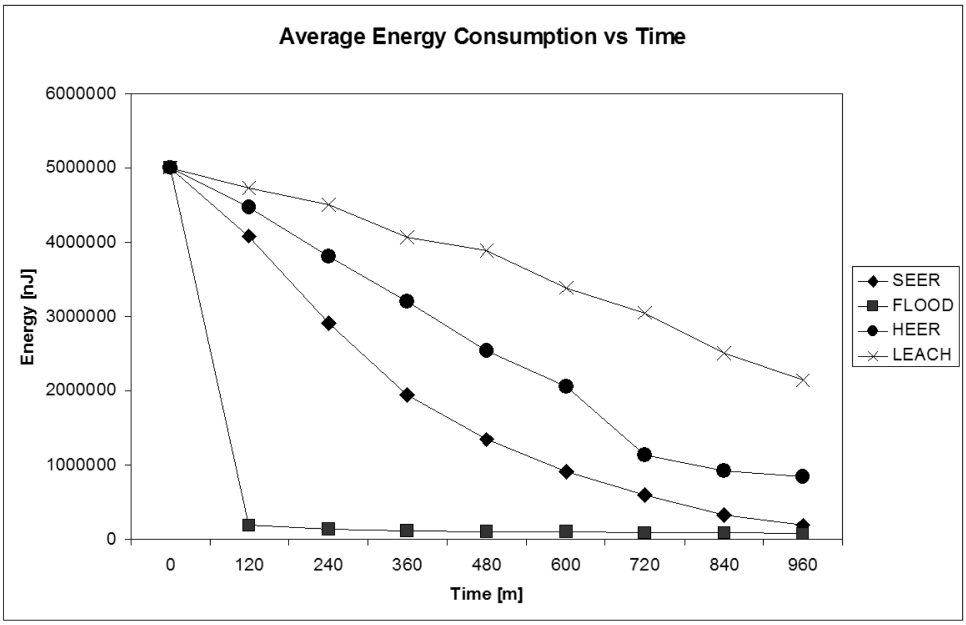

FIGURE 7.4: Average energy of the 50 node network over time.

7.2.5 Test 5 - *The average num. of messages sent in the network*:

As has been pointed out again and again, the number of transmissions that a network emits is the greatest consumer of the energy resource. These transmissions include any message traffic, i.e. ACK messages. In *Fig.* 7.5, it can be seen that all the nodes die quickly in the Flooding network. It is due to all the inactive nodes that the messages no longer increase. On the other hand, LEACH shows the number of tranmissions that are required for its' operation. The sharp increases are evident of the voting process taking place.

SEER's performance makes use of the least number of transmissions. For a marginal increase in the number of messages, HEER provides reliable transmission of critical messages. The increase at 600 minutes on HEER's graph shows evidence of the nodes adjusting their tree structure for more efficient routing.

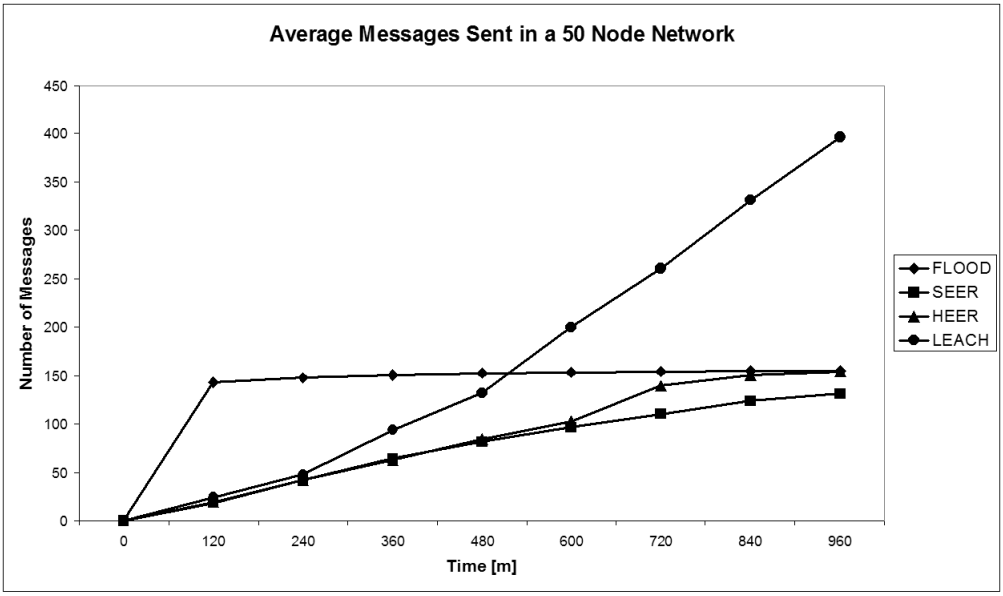

FIGURE 7.5: Average number of messages sent in a 50 node network.

7.2.6 Test 6 - *The num. of data messages received by the sink*:

Arguably the most important metric is the number of messages that the sink node receives, (for example, how many temperature readings have been recorded). The network size is varied to show how the number of messages is dependent on the sink neighbour nodes. After 45 minutes, each node will have sent two data messages, at 15 and 30 minutes respectively. The far right bar (100%) shows the number of message that should have been received by the sink node. We can see that all the protocols manage to route their messages for network sizes of 25 and 50. At the 100 node mark, Flooding begins to fail. HEER is the only protocol to route all 1998 messages in a 1000 node network. For larger network sizes, LEACH is able to transmit the most messages to the sink node, due to direct communcation. The biggest reason for some messages not reaching the sink node, is of course the now inactive sink neighbour nodes.

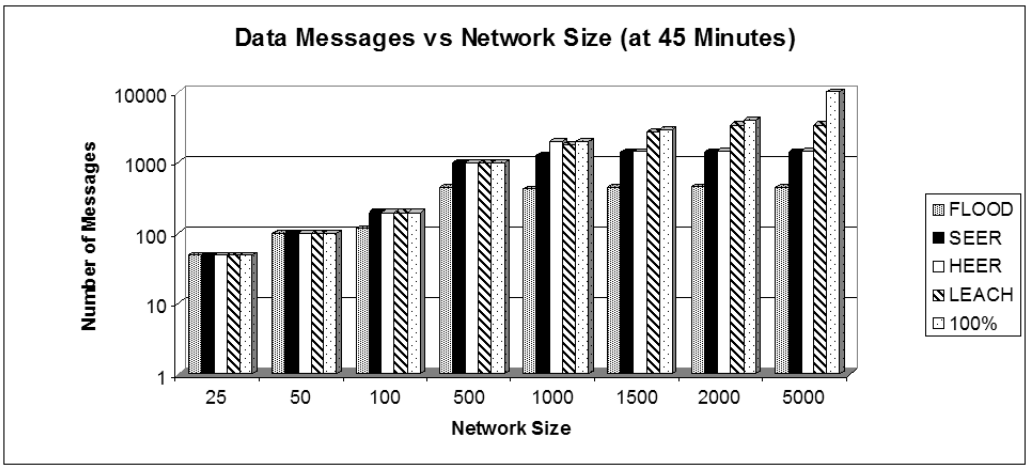

FIGURE 7.6: Number of data messages received by the sink node after 45 minutes.

7.3 SIMULATION RESULTS - RANDOM NETWORK

A uniform network, is not necessarily an indication of a typical network that can be found in a physical WSN environment. To this end some of the tests were conducted on a random network, an example is shown in *Fig.* 7.7 of a 50 node network. In the following figures of results, the term HEER-U refers to the uniform measures and HEER-R to the random.

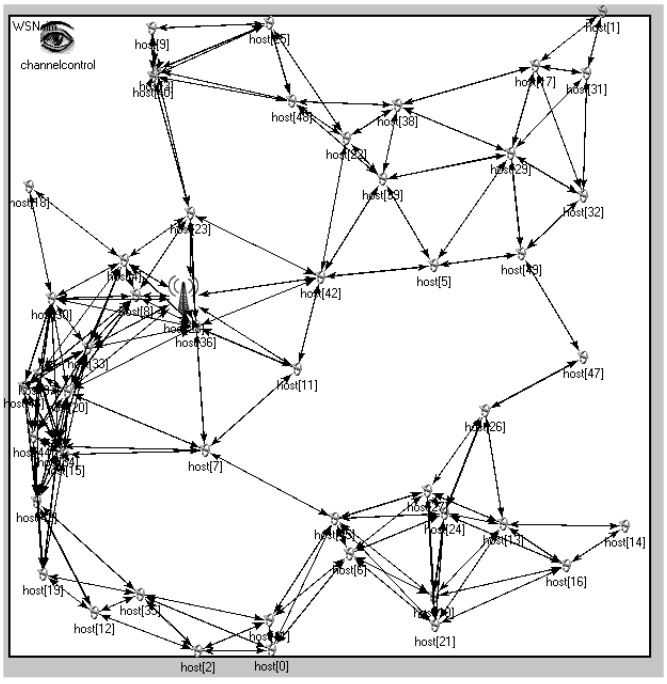

FIGURE 7.7: Example of the 50 node random network used in the following tests.

7.3.1 Test 1 - *Time until the first node dies*:

The number of connections between the nodes can be predetermined in the uniform network. In a random network more connections could possibly mean more messages. A routing protocol can also be specifically designed for a uniformed network, but it is the perforamance on a more realistic platform the provides a measure of a protocols' real world efficiency. In *Fig.* 7.8, we see the performance knock associated with more connections.

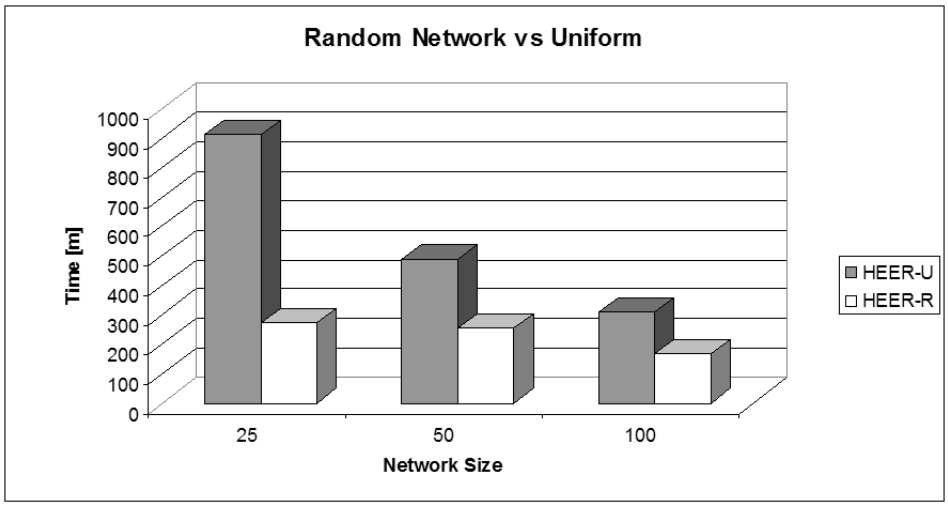

FIGURE 7.8: Time at which the first node fails due to depleting its energy source.

7.3.2 Test 4 - *The average remaining energy of the nodes*:

To better compare uniform results to those of a random network, the average energy was looked at *Fig.* 7.9. The random network had more sink neighbours than did the uniform network, thus, there were more shorter paths to the sink node. This allows HEER-R to outperform HEER-U.

7.3.3 Test 5 - *The average num. of messages sent in the network*:

In *Fig.* 7.10, a number of interesting occurances can be seen. Firstly HEER-U is able to route messages for longer along a path before an alternate needs to be found, this can be seen by the increase at 600 minutes. The random network, due to more neighbours and connections, is required to find alternate paths sooner, at 120 minutes for example.

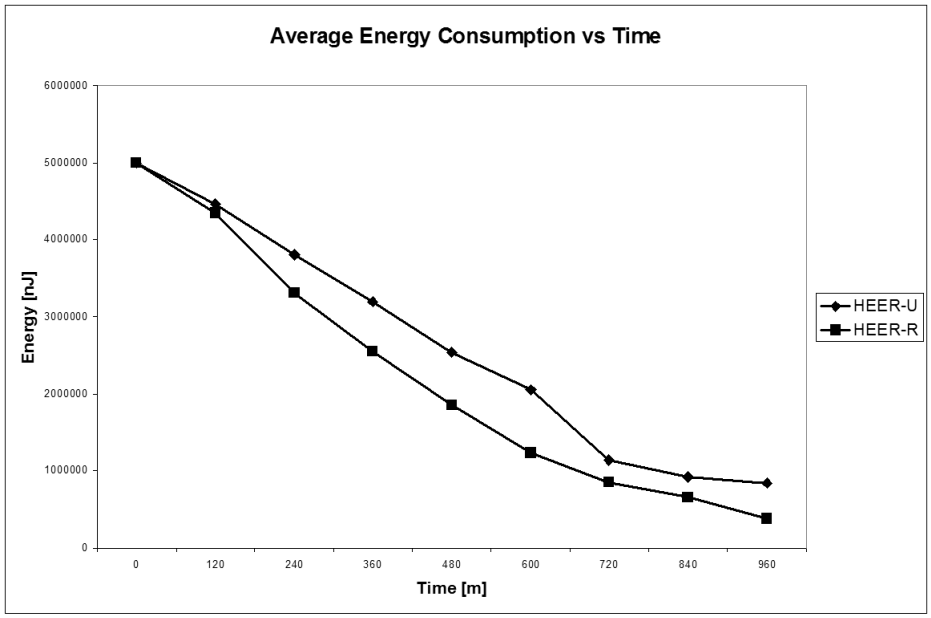

FIGURE 7.9: Average energy of the 50 node random network over time.

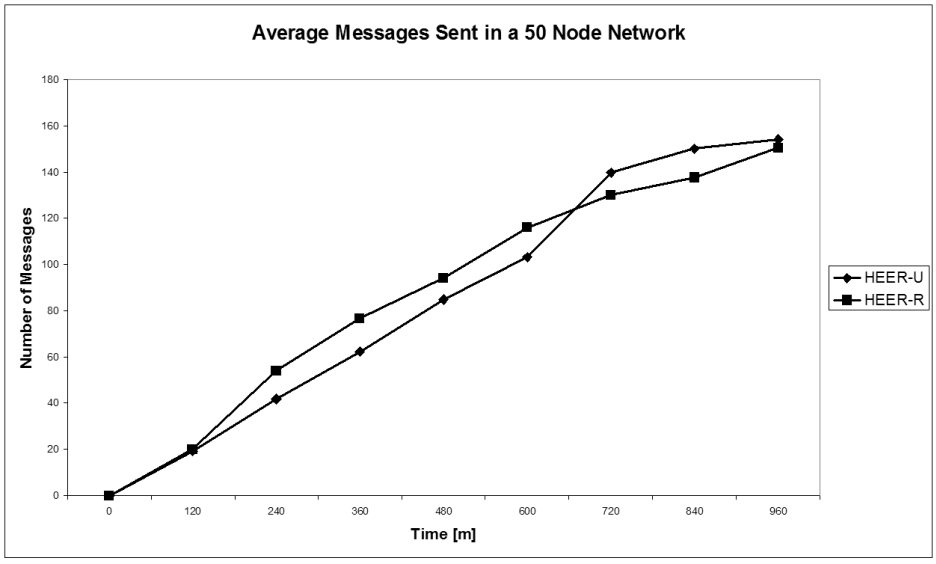

FIGURE 7.10: Average number of messages sent in a 50 node random network.

7.3.4 Effect of Variable Power Transmission

As has been mentioned before, the power required for transmission is varied in the implementation of HEER. The effects of this benefit will best be seen in a random network scenario where the distance between nodes is not equal. To see the results of this, a few simulations were conducted where the transmission power was fixed, similar to the other protocols. These results are shown in *Fig.* 7.11. The variable protocol performs in a more efficient manner when compared to the fixed power protocol. A protocol that makes use of variable transmission power will receive some benefits, but the advantage is more an add-on as opposed to a *"be all and end"* solution to the power problems faced by wireless sensor networks. Once again the main culprit is still the actual number of transmissions being sent and received.

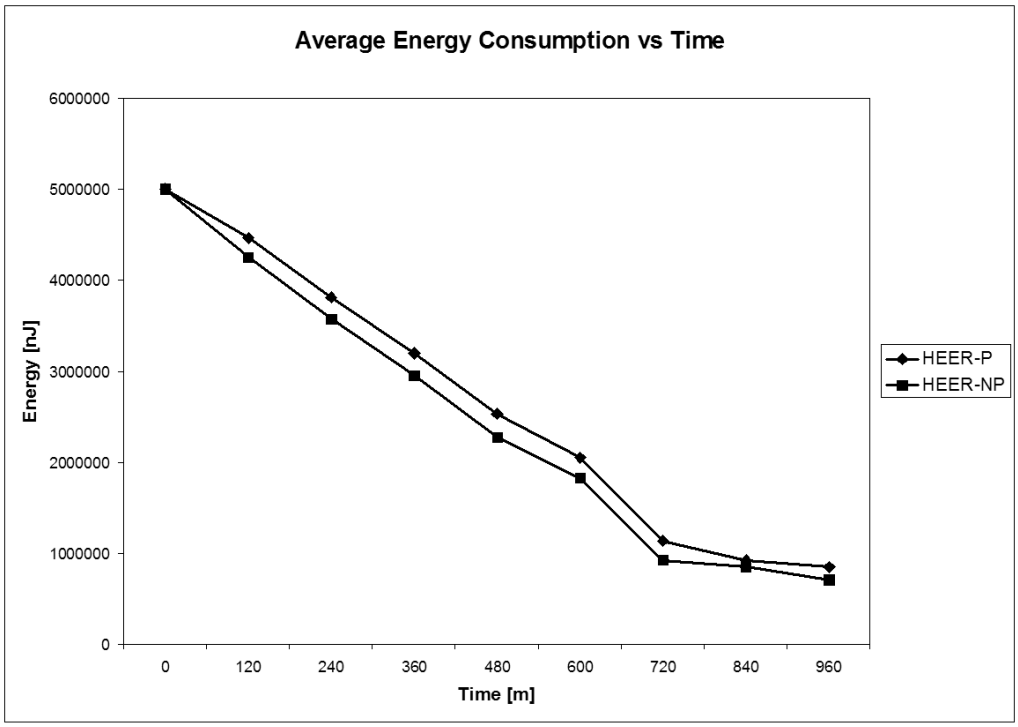

FIGURE 7.11: Comparision of variable and fixed power transmission.

7.4 CHAPTER SUMMARY

The results of HEER and the other simulated protocols were shown and briefly discussed. HEER held up well against SEER, outperforming it on a number of tests, while still providing data aggregation and reliable critical message delivery.

CHAPTER EIGHT
CONCLUSION

8.1 SUMMARY OF THE WORK

This research contributes a stable and adaptable routing protocol that is able to operate on any node hardware that meets the requirements for adaptable power transmission. The protocol functions on any network architecture and the simplicity and cross-layer design provides efficient energy use. The protocol is able to adapt to its environment and employ the routing architecture that would provide maximum results. This routing protocol may suffer from some of the disadvantages that are currently present in wireless sensor network protocols, but these have been minimised. The advantages of employing these features allow the protocol to meet other requirements of WSNs.

Simulator and protocol credibility was also raised. Many of the simulator products that are available show conflicting results, which could lead to a conclusion that they all may be incorrectly implemented. This means that the assumptions that they take into account when modeling the physical world are not sufficient [48]. A look into protocol validity, predominately that of SEER, was also conducted.

It is vital that the WSN research community start policing the research that is being conducted. Without this type of policy, the credibility of any research is brought into doubt.

A hybrid routing protocol, HEER, was designed. This protocol was built up from the humble beginnings of SEER [5]. The protocol introduces the following features:

- data aggregation,
- tree-like routing architecture,
- variable power transmission, and

- routing based on hopcount, “psuedo”-distance, and energy levels.

8.2 SUMMARY OF THE RESULTS

The designed hybrid protocol, HEER, was designed and implemented in a batch of simulations conducted on the Mobility Framework, operating on top of the OMNET++ simulator. The results obtained for SEER differ slightly from those shown in [5], but the reasons for this were discussed in Chapter 4.

HEER was shown to outperform SEER, and on occasion even a hierarchical protocol like LEACH. The results show that the hybrid approach, i.e. the tree structure, allows the protocol to make informed decisions about where to route a message. The results shown for SEER differentiate slightly from those shown in [5]. The main comparison has shifted to show SEER’s performance when compared to more realistic WSN protocols.

It can be concluded that HEER combines the elements of flat and hierarchical routing protocols, and is able to perform with ease in the demanding environment that is a WSN.

8.3 CRITICAL EVALUATION

The protocol designed in this document has met the design goal of being an energy efficient routing protocol. With the results having been confirmed via simulation, unfortunately does not confirm but merely gives an indication of the possible real world results, as discussed in chapter 4. Designing a protocol based on another, does allow a direct comparison between the protocols. Thereby validating the results to a limited (simulation only) scenario.

To further increase the credibility of the protocols simulated, requires a real world implementation on existing hardware. The problems with this approach, particularly in financially constrained circumstances, is difficult at best. The number of nodes to effectively test the protocol on large networks is the cause of the problem. Raising the credibility crisis being felt in the WSN environment has influenced the design of HEER. This knockon effect has improved the credibility of both SEER and HEER.

Although not all of the points raised in chapter 4 have been addressed, the majority of them have been seen too. By comparing this work to the points raised in section 4.3 the some of the more important aspects are listed below:

- Lack of independant repeatablility: All settings required to duplicate the simulations have been documented in this thesis. It should thus be possible for the simulations to be run and similar results to be achieved.
- Lack of statistical validity: In all simulations where random variables were used, the seed for the pseudorandom number generators was altered each time. One of the most important factors for the simulations was to ensure that each run was independant from the other.
- Improper/nonexistant validation: A real world comparison was not possible, so the competitive protocols were chosen such that they are well known, implementation is well documented, and comparative results are available.
- Unrealistic application traffic: A generic application was chosen for the simulations, although this may not have been the best choice. For a routing protocol to remain as generic as possible with the ability to cater for as many application types as possible was the main driving force behind this decision.

Criticism of any work is a difficult task but one that only further benefits the research field.

8.4 FUTURE WORK

The research proposed in this work is not the be all and end all of WSN research. The following list gives a few ideas for future work. The possiblities are endless and are only bounded by human thought. Although in reality, the possiblities are and will be bounded by economics and relevant applications.

- HEER could be expanded or re-worked to include mobility features, as well as add the necessary functionality to allow it to operate in a WSAN environment.

- The supporting lower layers, PHY and MAC, should be designed for HEER. An investigation into implementing a TDMA MAC protocol for HEER should be conducted. This could lead to more energy savings.

- As has been pointed out previously, simulators for WSNs leave the credibility of their results hanging. Further study into producing either a new simulator or adjusting current ones to better allow for the unique requirements of a WSN should be investigated. A specific framework could also be proposed. This framework would allow all future simulator results to conform to a more rigid standard. This may bring some of the credibility and validation back to the results to be presented in future research.

- Any protocol or algorithm designed specifically for a single application will out-perform general-purpose protocols. The problem with this is that each application possibly requires the design, development and implementation of its own propreity protocol. A framework should be designed that allows for a plug-and-play attitude to protocol expandability. Small modules could be written that allow for functionality to be added when needed for an application, i.e. localization.

- South Africa has vast natural resources that are mined beneath the surface. This unique underground environment provides challenges that a WSN would need to overcome. These challenges are across the board, starting from the physical all the way to the application layer. The unique challenge of designing a node and its respective protocols would allow for a product to fulfill this niche market need.

REFERENCES

[1] *The Internet of Things*, 1st ed. International Telecommunication Union, Geneva, November 2005, Last accessed: Jul. 2006. [Online]. Available: www.itu.int/internetofthings/

[2] K. Pister. (2001) Smartdust: Autonomous Sensing and Communication in a Cubic Millimeter. Last accessed: Oct. 2006. [Online]. Available: http://robotics.eecs.berkeley.edu/~pister/SmartDust/

[3] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE Wireless Communications and Networking*, vol. 7, no. 5, pp. 16–27, October 2000.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks (Elsevier)*, vol. 38, pp. 393–422, 2002.

[5] C. Leuschner, "The Design of a Simple Energy Efficient Routing Protocol to Improve Wireless Sensor Network Lifetime," Master's thesis, Electrical, Electronic and Computer Engineering, University of Pretoria, April 2005. [Online]. Available: http://upetd.up.ac.za/thesis/available/etd-01242006-091709/unrestricted/00dissertation.pdf

[6] J. Al-Karaki and A. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, December 2004.

[7] G. Hoblos, M. Staroswiecki, and A. Aitouche, "Optimal Design of Fault Tolerant Sensor Networks," in *Proceedings of the IEEE International Conference on Control Applications, September 2000, Anchorage, USA*, 2000, pp. 467–472.

[8] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, "Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems," in *Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA), July 2001, Ambleside, UK*, 2001, pp. 1–6.

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy Efficient Communication Protocol for Wireless Microsensor Networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS)*, January 2000, pp. 1–10.

[10] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, USA*, 2002, pp. 88–97.

[11] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks," in *Proceedings of the International Conference on Mobile Computing and Networking (MOBICOM), 15-19 August 2001, Rome, Italy*, 2001, pp. 272–287.

[12] I. Akyildiz and I. Kasimoglu, "Wireless Sensor and Actor Networks: Research Challenges," *Ad Hoc Networks (Elsevier)*, vol. 2, pp. 351–367, 2004.

[13] V. Bahl. (2002) Zigbee Abstract. Last accessed: Nov. 2004. [Online]. Available: http://www.zigbee/resources

[14] N. Gunasekaran, S. Rama-Reddy, and K. Sairam, "Bluetooth in Wireless Communication," *IEEE Communications*, pp. 90–96, June 2002.

[15] J. Haartsen and S. Mattisson, "A New Low-power Radio Interface Providing Short-range Connectivity," *Proceedings of the IEEE*, vol. 88, no. 10, pp. 1651–1661, 2000.

[16] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LRWPANS). Std 802.15.4*. IEEE Computer Society, 2003.

[17] B. Heile. (2004) CES 2004 Presentation. Last accessed: Nov. 2004. [Online]. Available: http://www.zigbee/resources

[18] C. Evens-Pughe, "Bzzz: Is the ZigBee Wireless Standard, promoted by an alliance of 25 firms, a big threat to Bluetooth?" *IEE Review*, pp. 28–31, March 2003.

[19] V. Bahl. (2003) Zigbee Tutorial. Last accessed: Nov. 2004. [Online]. Available: http://www.zigbee/resources

[20] P. Kinney. (2003) Zigbee Technology: Wireless Control that Simply Works. Last accessed: May 2005. [Online]. Available: http://www.hometoys.com/htinews/oct03/articles/kinney/zigbee.htm

[21] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Std 802.11,*. IEEE Computer Society, 1999, Last accessed: Apr. 2005.

[22] J. Chen, C. Lee, and J. Yeh, "WLAN Standards," *IEEE Potentials*, pp. 16–22, October 2003.

[23] B. Allen, M. Dohler, E. Okon, W. Malik, A. Brown, and D. Edwards, *Ultra-wideband Antennas and Propagation for Communications, Radar and Imaging*. John Wiley & Sons, 2007.

[24] L. Paulson, "Will Ultrawideband Technology Connect in the Marketplace?" *Computer*, vol. 36, no. 12, pp. 15–17, December 2003.

[25] (2006) Ultra-Wide Band. Last accessed: Oct. 2006. [Online]. Available: http://en.wikipedia.org/wiki/Ultra-wideband

[26] *Wibree*. Nokia Corporation, 2006, Last accessed: Oct. 2006. [Online]. Available: http://www.wibree.com/

[27] S. Hedetniemi and A. Liestman, "A Survey of Gossiping and Broadcasting in Communication Networks," *IEEE Network*, vol. 18, no. 4, pp. 319–349, 1988.

[28] W. Heinzelmann, K. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," in *Proceedings of the 5th International Conference on Mobile Computing and Networking (Mobicom), 15-19 August 1999, Seattle, USA*, 1999, pp. 174–185.

[29] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of the 6th International Conference on Mobile Computing and Networking (Mobicom), 6-11 August 2000, Atlanta, USA*, 2000, pp. 56–67.

[30] D. Braginsky and D. Estrin, "Rumour Routing Algorithm for Sensor Networks," in *Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), October 2002, Atlanta, Georgia, USA*, 2002, pp. 22–31.

[31] R. Shah and K. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 17-21 March, 2002, Orlando, Florida, USA*, vol. 1, 2002, pp. 350–355.

[32] K. Matrouk and B. Landfeldt, "Energy-Conservation Clustering Protocol based on Heat Conductivity for Wireless Sensor Networks," in *Proceedings of the Intelligent Sensors, Sensor Networks and Information Processing Conference (ISSNIP 04), 14-17 December 2004*, 2004, pp. 19–24.

[33] A. Manjeshwar and D. Agrawal, "TEEN: A Protocol for Enhanced Efficiency in Wireless Sensor Networks," in *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, April 2001, San Francisco, California, USA*, 2001.

[34] D. Agrawal and A. Manjeshwar, "APTEEN: A Hybrid Protocol for Enhanced Efficiency in Wireless Sensor Networks," in *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS 02)*, 2002, pp. 195–202.

[35] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, 1st ed. John Wiley & Sons, 2005.

[36] Y. Liu, C. Li, and C. Cruz, "Performance Optimization for a Mobile Small-Unit-Operation Ssituational Awareness (suo-sas) Radio Network," Presented at the IEEE Military Communications Conference (MILCOM 03), October 2003, 2003.

[37] D. Capyioni and A. Russo, "Small Unit Operations Situation Awareness System (suo-sas) Radio Architecture and System Field Testing Results," Presented at the IEEE Military Communications Conference (MILCOM 03), October 2003, 2003.

[38] V. Kawadia and P. Kumar, "A Cautionary Perspective on Cross-Layer Design," *IEEE Wireless Communication Magazine*, vol. 12, no. 1, pp. 3–11, February 2005.

[39] M. Conti, S. Giordano, G. Maselli, and G. Turi, "Cross-Layering in Mobile Ad-Hoc Network Design," *IEEE Computer, Special Issue on AdHoc Networks*, vol. 37, no. 2, pp. 48–51, 2004.

[40] B. Sadler, "Fundamentals of Energy-Constrained Sensor Network Systems," *IEEE A&E Systems Magazine*, vol. 20, no. 8, pp. 17–35, August 2005.

[41] D. Cavin, Y. Sasson, and A. Schiper, "On the Accuracy of MANET Simulators," in *Proceedings of the Workshop on Principles of Mobile Computing (PMOC'02), 30-31 October 2002, Toulouse, France*, 2002, pp. 38–43.

[42] A. Varga. (2005, March) Omnet++ Discrete Event Simulator System User Manual. Last accessed: Oct. 2006. [Online]. Available: http://www.omnetpp.org/doc/manual/usman.html

[43] "OMNeT++ Discreet Event Simulation System," 2006, Last accessed: Nov 2006. [Online]. Available: http://www.omnetpp.org

[44] "Mobility Framework for OMNeT++," 2006, Last accessed: Nov. 2006. [Online]. Available: http://mobility-fw.sourceforge.net

[45] M. Takai, J. Martin, and R. Bagrodia, "Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks," in *MobiHoc 2001*, 2001.

[46] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K. Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan, "Effects of Detail in Wireless Network Simulation," in *Proceedings of the SCS Multiconference on Distributed Simulation, January 2001, Phoenix, USA*, 2001, pp. 3–11.

[47] S. Kurkowski, T. Camp, and M. Colagrosso, "Manet Simulations Studies: The Incredibles," *SIGMobile Mobile Computing Comm. Rev.*, vol. 9, no. 4, pp. 50–61, 2005.

[48] T. Andel and A. Yasinsac, "On the Credibility of Manet Simulations," *Computer*, vol. 39, no. 7, pp. 48–54, July 2006.

[49] R. Jain, *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, 1999.

[50] J. Proakis and M. Salehi, *Communications Systems Engineering*, 2nd ed. Prentice Hall, 2002.

[51] M. D. Renzo, F. Graziosi, R. Minutolo, M. Mantanari, and F. Santucci, "The Ultra-wide Bandwidth Outdoor Channel: From Measurement Campaign to Statistical Modelling," *Mobile Networks and Applications*, vol. 11, no. 4, pp. 451–467, August 2006.

[52] "CC1000 Single Chip Low Cost Low Power RF Transceiver," 2006, Last accessed: Nov 2006. [Online]. Available: http://www.Chipcon.com

Contact Information

Postal Address	794 Thomas Avenue
	Arcadia
	Pretoria
	0083
Email	jgpage@ieee.org