THE DESIGN OF A SIMPLE ENERGY EFFICIENT ROUTING PROTOCOL TO IMPROVE WIRELESS SENSOR NETWORK LIFETIME

by

Charl Jaco Leuschner

Submitted in partial fulfilment of the requirements for the degree
Master of Engineering (Computer Engineering)
in the
Faculty of Engineering, the Built Environment and Information Technology
UNIVERSITY OF PRETORIA

April 2005

ACKNOWLEDGEMENTS

A special word of thanks to my supervisor, Prof Gerhard Hancke of the Department of Electrical, Electronic & Computer Engineering at the University of Pretoria for his continued support and guidance throughout this research. I would also like to thank him for providing me with the opportunity to visit the Vienna University of Technology where I conducted a large part of the research.

I would like to thank Prof Dietmar Dietrich and Dr Stefan Mahlknecht of the Institute of Computer Technology at the Vienna University of Technology for providing guidance and advice during my research there.

Special thanks to my family and my fiancée, Eunice, who supported and motivated me continuously. I would not have been able to complete this research if it was not for them.

Most important of all: thanks to God, who deserves all the credit. For nothing is possible without Him.

The Design of a Simple Energy Efficient Routing Protocol to Improve Wireless Sensor Network Lifetime

by

Charl Jaco Leuschner

Supervisor:	Prof G.P. Hancke
Department:	Electrical, Electronic and Computer Engineering
Degree:	Master of Engineering (Computer Engineering)

Key terms

hierarchical routing; flat routing; wireless sensor networks; source initiated; destination initiated; energy consumption; energy efficiency; node lifetime; clustering; distributed sensing

Summary

The number of potential applications for wireless sensor networks is immense. These networks may consist of large numbers of low cost, low power, disposable sensor nodes that can be deployed inside or close to phenomena to be monitored. The nature of these networks necessitates specific design requirements, of which energy efficiency is paramount.

The limited available energy of sensor nodes is mainly drained during communication and computational processing. An energy efficient routing protocol can limit the number of message transmissions and the computational complexity of finding routing paths. Many routing protocols have been proposed for wireless sensor networks. Most of them are computationally complex, require a large number of messages to be transmitted or require that sensor nodes possess certain hardware capabilities in order to function.

The objective of this dissertation was to develop a Simple Energy Efficient Routing (SEER) protocol for wireless sensor networks that is computationally simple, reduces the number of transmitted messages and does not impose any hardware prerequisites.

The new routing protocol, which was developed during this research, uses a flat network structure for scalability and source initiated communication along with event-driven reporting to reduce the number of message transmissions. Computational simplicity is achieved by using a simple method for routing path selection.

The SEER protocol selects the next hop for a message by choosing a neighbour that has a smaller or equal hop count to the current node. If multiple neighbours satisfy this requirement, the neighbour with the highest remaining energy is chosen as the next hop. Each node in the network has a table containing the hop count and remaining energy of each of its neighbours. Periodic messages sent through the network update these neighbour tables.

SEER uses a novel approach to select the next hop of a message during routing. The protocol increases the lifetime of the network dramatically, compared to other similar routing protocols. This improvement is directly related to the reduction in the number of transmissions made by each node. The simplicity of the protocol reduces the required computational processing compared to other protocols, and at the same time makes this one of the few available protocols that does not impose hardware requirements on nodes in order to function.

Die Ontwerp van 'n Eenvoudige Energie-Effektiewe Roeteringsprotokol om die Leeftyd van Koordlose Sensor Netwerke te Verbeter

deur

Charl Jaco Leuschner

Studieleier: Prof. G.P. Hancke
Departement: Elektriese, Elektroniese en Rekenaar-Ingenieurswese
Graad: Meester van Ingenieurswese (Rekenaar-Ingenieurswese)

Sleutelterme

hiërargiese roetering; plat roetering; koordlose sensornetwerke; bron-geïnisieer; bestemming-geïnisieer; energieverbruik; energie-effektiwiteit; node-leeftyd; groepering; verspreide waarneming

Opsomming

Daar bestaan 'n ontsaglike hoeveelheid moontlike toepassings vir koordlose sensornetwerke. Hierdie netwerke bestaan uit 'n groot hoeveelheid lae koste, lae drywing, weggooibare sensor nodusse wat naby of binne-in verskynsels wat waargeneem moet word, ontplooi kan word. Die aard van hierdie tipe netwerke veroorsaak spesifieke ontwerpsvereistes, waarvan energie-effektiwiteit die belangrikste is.

Die beperkte beskikbare energie van sensornodusse word hoofsaaklik deur kommunikasie en rekenkundige verwerking gedreineer. 'n Energie-effektiewe roeteringsprotokol kan die hoeveelheid boodskappe wat gestuur word, sowel as die kompleksiteit van die bewerkings om roetes te vind, minimeer. Verskeie roeteringsprotokolle is vir koordlose sensornetwerke voorgestel. Die meeste van hulle is rekenkundig kompleks, vereis dat 'n groot hoeveelheid boodskappe gestuur moet word of vereis dat sensornodusse oor sekere hardeware vermoëns beskik vir die protokol om te funksioneer.

Die doelwit van hierdie verhandeling is om ’n Eenvoudige Energie-Effektiewe Roeteringsprotokol (“SEER”) vir koordlose sensornetwerke te ontwikkel wat rekenkundig eenvoudig is, die hoeveelheid boodskappe wat gestuur word verminder en geen hardeware voorvereistes het nie.

Die nuwe roeteringsprotokol, wat gedurende hierdie navorsing ontwikkel is, gebruik ’n plat netwerk struktuur om dit bruikbaar te maak vir netwerke van enige grootte en ’n kombinasie van bron geïnisieerde kommunikasie en gebeurtenis-gedrewe verslaggewing om die hoeveelheid boodskappe wat gestuur word te verminder. Rekenkundige eenvoud word bewerkstellig deur ’n eenvoudige metode te gebruik om die roeteringspad te kies.

Die SEER protokol kies die volgende “hop” vir ’n boodskap deur ’n aangrensende nodus te kies wat ’n kleiner of gelyke hop-telling as die huidige nodus het. Indien meer as een aangrensende nodus aan hierdie vereiste voldoen, word die aangrensende nodus met die hoogste oorblywende energie as die volgende hop gekies. Elke nodus in die netwerk beskik oor ’n tabel met die hop-telling en oorblywende energie van elk van sy aangrensende nodusse. Hierdie tabelle word opgedateer deur periodiese boodskappe wat deur die netwerk gestuur word.

SEER gebruik ’n doeltreffende benadering om die volgende hop van ’n boodskap gedurende roetering te kies. Die protokol verhoog die leeftyd van die netwerk dramaties in vergelyking met ander soortgelyke protokolle. Hierdie verbetering hou direk verband met die vermindering van die hoeveelheid boodskappe wat elke nodus hoef te stuur. Die eenvoud van die protokol verlaag die vereiste rekenkundige prosessering en terselfdertyd maak dit hierdie protokol een van die enkeles wat geen hardeware voorvereistes vir nodusse inhou om te kan funksioneer nie.

List of Abbreviations

ADC	analogue to digital converter
AES	advanced encryption standard
AODV	ad hoc on demand distance vector
BATR	balanced aggregation tree routing
CPU	central processing unit
CSMA	carrier sense multiple access
CSMA-CA	carrier sense multiple access with collision avoidance
DARPA	Defense Advanced Research Projects Agency
DSN	distributed sensor networks
DSSS	direct sequence spread spectrum
FFD	full function device
FHSS	frequency-hopping spread spectrum
GBR	gradient-based routing
GHz	gigahertz
GPS	global positioning system
HAR	hierarchy-based anycast routing
HEAR-SN	hierarchical energy-aware routing for sensor networks
IEEE	Institute of Electrical and Electronic Engineers
IP	Internet protocol
kB	kilobyte
kbps	kilobits per second
L2CAP	logical link control and adaptation protocol
LAN	local area network
LEACH	low energy adaptive clustering hierarchy
LMP	link manager protocol

MAC	medium access control
mAh	milliamp hour
Mbps	megabits per second
MCFA	minimum cost forwarding algorithm
MECN	minimum energy communication network
MEMS	micro electro-mechanical systems
MHz	megahertz
MIT	Massachusetts Institute of Technology
PHY	physical
RAM	random access memory
RFD	reduced function device
RISC	reduced instruction set computer
SEER	simple energy efficient routing
SMECN	small minimum energy communication network
SOSUS	sound surveillance system
SPIN	sensor protocols for information via negotiation
TDMA	time division multiple access
TEDS	transducer electronic data sheets
TTL	time to live
TUV	Vienna University of Technology
UCLA	University of California, Los Angeles
UWB	ultrawideband
WANET	wireless ad hoc network
WINS	wireless integrated network sensors
WLAN	wireless local area network
WPAN	wireless personal area network
WSN	wireless sensor network
WSSN	wireless self-sustaining sensor network

Table of Contents

CHAPTER 1

Introduction

In 1999 it was called one of *“21 ideas for the 21st century”* [1] and in 2003 it was heralded as one of *“10 emerging technologies that will change the world”* [2]. This revolutionary technology is known as *Wireless Sensor Networks (WSNs)*. Such a sensor network is deployed in or close to the phenomenon that it has to monitor. Applications range from environmental monitoring to industrial sensing to military applications and far beyond; possibly only limited by human imagination. The development of this technology has been fuelled by advances in electronic miniaturisation (including micro electro-mechanical systems (MEMS) technology), wireless communication and low-cost manufacturing.

Small, inexpensive, low-power, intelligent, disposable sensors can be deployed in large numbers, in environments ranging from the home to hostile and possibly inaccessible environments such as disaster areas or battlefields. They can be deployed manually by hand, or randomly by, for instance, dropping them from an aeroplane. These sensor nodes are self-configuring and contain one or more sensors, embedded wireless communications and data processing components and a limited energy source. Due to the large number of nodes and the possibly hazardous environment in which these nodes are deployed, their batteries cannot be replaced. The failure of a single node in the network could possibly cause network partition and cut a part of the WSN off from the rest of the network. Network lifetime is therefore dependent on the lifetime of individual nodes. This necessitates energy efficient design on every layer of the protocol stack. In support of this requirement for energy efficiency, this research focuses on the network layer and, more specifically, the development of an energy efficient routing protocol.

In addition to the possibly very large number of sensing nodes, a WSN may also have a base station. The sensing nodes have to route data about their environment to the base station. A sensing node is sometimes called a *source* and a base station is sometimes called a *sink*. The sink node collects and interprets the data from all the source nodes in the network. The sink node may be connected to a wired network and may not have an energy limitation. The source nodes, on the other hand, are dependent on their limited batteries and are removed from the network when their batteries are depleted.

1.1 Scope

The scope of this research is the development of an energy efficient routing protocol for WSNs that can be easily implemented on existing WSN nodes.

Even though there are numerous proposals for WSN routing protocols, there is still a great need for new protocols that can extend network lifetime, can be implemented easily on nodes using current technology and can be used for networks of any size.

1.2 Motivation

Wireless Sensor Networks are an emerging technology that will be used in more and more applications as the state of the art in communication, sensing and energy technologies advance. The main design challenge of WSN nodes is energy efficiency on every layer of the protocol stack. According to [3] the battery energy of a node is depleted by: (i) computational processing and (ii) transmission and reception of data. Both of these factors are controlled by the network layer. An energy efficient network layer can reduce the number of messages that are sent by a node as well as the complexity of the computation of routing paths, thereby maximizing node lifetime.

Routing protocols specifically designed for WSNs, first appeared in the literature in the late 1990’s. The specific restrictions imposed by a WSN require special routing protocols to be used. WSN routing protocols can be classified based on the network structure, as either *flat* or *hierarchical*. In flat networks all of the nodes are equal and can participate equally in the routing task. Hierarchical networks on the other hand, require some nodes to control the communication of other nodes. Flat protocols are more efficient for use in WSNs than hierarchical protocols due to the fact that they are more scalable, generally require fewer messages to be sent and are simpler in computational requirements.

Routing protocols can be further subdivided into *source initiated* and *destination initiated* protocols. Nodes employing source initiated protocols send data either periodically, or in response to certain events in their environment. In destination initiated routing, nodes only send data in response to a request for data. The drawback of destination initiated protocols is the fact that requests are usually flooded through the network, draining the energy sources of nodes. Therefore, source initiated operation can achieve higher energy efficiency.

The flat routing protocols that have been proposed in the literature generally either require an excessive number of messages for data transfer ([4], [5], [6], [7], [8]), are computationally complex [9], require sensor nodes to have certain physical capabilities ([10], [9], [11]) or require a certain network structure ([10], [11], [12]) for the protocol to function.

The problem addressed by this research is the lack of a routing protocol for WSNs that is energy efficient, can be used in a general sensor network and does not impose hardware requirements.

1.3 Objectives

The objective of this research is to *develop a new WSN routing protocol* that has the following characteristics:

1. The protocol must be scalable and function efficiently for networks of any size.
2. The protocol has to minimise the number of transmissions made by a WSN node.
3. The protocol has to minimize the computational processing that the nodes have to perform during routing.
4. The protocol must be computationally simple and simple to implement.
5. The protocol must not depend on hardware capabilities of nodes.

The objectives for the routing protocol to be developed can be summarized as: *scalability, energy efficiency, simplicity* and *practicality*.

1.4 Contribution

This research has two very important contributions to the body of knowledge in the field of WSN routing protocols. The first is the fact that the protocol developed during this research, named *Simple Energy Efficient Routing (SEER)*, can be implemented without modification, on currently existing nodes and future nodes, since there are no hardware pre-requisites.

The second important contribution is the simplicity of the protocol. Many hierarchical protocols require complex network setup procedures in order to build routing trees or choose cluster heads. Most of the destination-initiated routing protocols require that multiple messages be exchanged in order to request data. Source-initiated protocols also require their own message exchanges in order to advertise available data. Many protocols also make assumptions, such as that all the nodes in the network are time synchronised or all of the nodes are one hop from the sink, which limits their ability to be implemented. This research minimizes assumptions and computational complexity in order to produce a useful and practical protocol.

The scalability achieved by this protocol is a significant contribution to the field of study. Most protocols simulate networks of up to a hundred nodes to verify their designs and a few multi-hop protocols use up to six hundred nodes. One or two protocols use up to two thousand nodes in their results, but one such protocol assumes that all of the sensor nodes are one hop from the sink. This research contributes to the small knowledge base of simulation results pertaining to large networks.

1.5 Research Methodology

A key component to the design of any routing protocol is a thorough knowledge and understanding of the factors that influence the specific network for which the routing protocol is intended. Therefore, a thorough literature study was done to identify and investigate the factors that influence the design of WSN routing protocols. The literature study also includes an investigation into available WSN routing protocols, in order to identify common problems faced by these protocols. A new protocol was designed, taking into consideration the specific requirements of WSNs and the common flaws of available protocols. The developed protocol was simulated to verify its functionality and compared, in simulation, against other available protocols, to verify its improvement in network lifetime over these protocols.

1.6 Overview

Chapter 2 provides an introduction to wireless sensor networks. It discusses the evolution of WSNs, the differences between WSNs and traditional ad hoc wireless networks, possible WSN applications, factors that influence the design of WSNs and concludes with a discussion of emerging standards in WSNs.

WSN routing protocols are investigated in Chapter 3. The challenges and design issues related to WSN routing are discussed and a classification of WSN protocols follows. A thorough overview of some of the available protocols is then given and the chapter ends with a comparison of these.

In Chapter 4, the protocol designed during this research is described. The chapter starts off with a discussion of the design goals of the protocol and how the goals were achieved. The chapter then gives an explanation of the operation of the protocol.

Comprehensive simulation results and an analysis of each set of results are presented in Chapter 5.

This research is concluded in Chapter 6.

CHAPTER 2

Wireless Sensor Networks

2.1 WSN Evolution

Sensor network development was initiated by the United States during the Cold War [13]. A network of acoustic sensors was placed at strategic locations on the bottom of the ocean to detect and track Soviet submarines. This system of acoustic sensors was called the Sound Surveillance System (SOSUS). During the same time period the United States also deployed networks of radars for air defence. These sensor networks used hierarchical processing, where data is processed at different layers until the data of interest reaches the user. Human operators played an important role in these systems. Both of these sensor networks were wired networks that did not have the energy or bandwidth constraints of wireless systems are two of the main design issues related to WSN routing protocols.

Modern sensor network research, however, started in the early 1980's at the Defense Advanced Research Projects Agency (DARPA) in the United States [13]. The Distributed Sensor Networks (DSN) program, as it was known, assumed a network with many independent, low-cost sensing nodes that were spatially distributed but able to collaborate. Information was routed to the node that could use it best. In the mid 1980's the Massachusetts Institute of Technology (MIT) developed a demonstration DSN consisting of acoustic sensors designed to track low-flying aircraft [13]. Microphones, arranged in arrays of six, were used for the acoustic sensing. The mobile vehicle nodes consisted of one computer and three processors, with 256kB memory and 512kB shared memory, processing the acoustic signals [13]. Energy was supplied by an acoustically quiet generator, mounted on the back of the vehicle node. The nodes communicated with microwave radio and Ethernet was used for fixed line communication. One of the mobile vehicle nodes is shown in Figure 2.1 and the equipment rack of the node is shown in Figure 2.2.

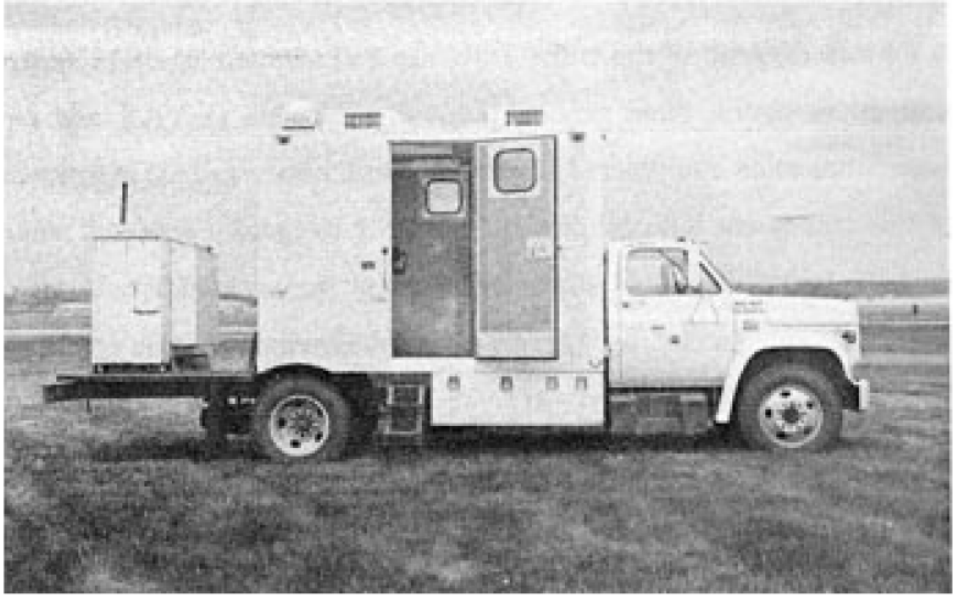

Figure 2.1: MIT mobile vehicle DSN node [13].

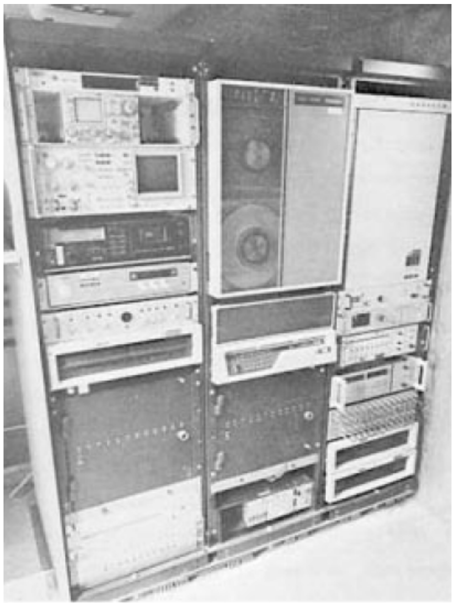

Figure 2.2: Equipment rack of an MIT mobile vehicle DSN node [13].

Wireless sensor networks have evolved immensely since research into DSNs started in the early 1980's. One of the more recent WSN projects is the Wireless Self-Sustaining Sensor

Network (WSSN) project of the Institute of Computer Technology at the Vienna University of Technology (TUV) [14]. The research focused on the development of low-cost, energy efficient hardware and an energy efficient medium access control (MAC) protocol [15]. One of the nodes that was developed is shown in Figure 2.3. The three most important aspects of any sensor node are its microcontroller, wireless interface and energy source. Each of the WSSN nodes has a 16 bit, 4MHz RISC CPU with 8kB of flash memory and 256 bytes of RAM. Their wireless interface consists of a 1Mbps, 2.4GHz transceiver. They come equipped with a temperature sensor with ±0.5°C accuracy and a 10 bit analog interface for an optional sensor. The energy conservation techniques employed by these nodes is very noteworthy.

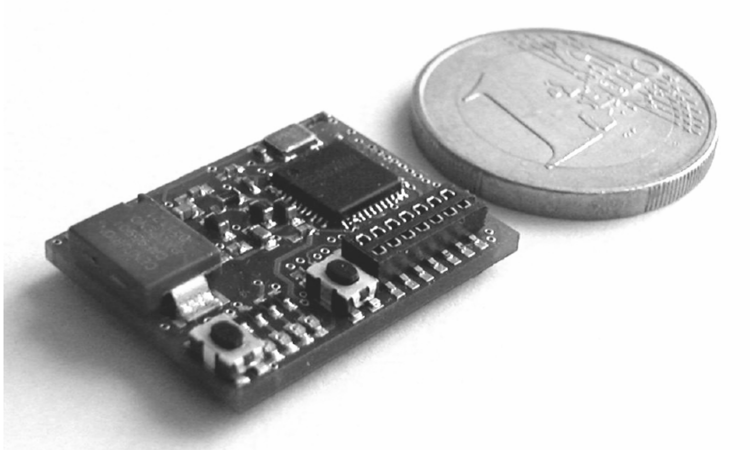

Figure 2.3: A TUV WSSN node is as small as one Euro [14].

These nodes consume about 100μW of power on average if certain environmental conditions are assumed, as well as that each node receives and transmits 120 bits of data every five seconds. These nodes would be able to operate for almost nine years, if each node was equipped with a 3200mAh lithium battery that has an input voltage of 2V. This lifetime is achieved by using state of the art energy storage as well as energy scavenging. For storage, WSSN nodes use a combination of ultracapacitors and lithium accumulators. The advantage of ultracapacitors is the fact that they can quickly absorb large amounts of energy, but on the downside, their leakage current grows exponentially as the applied voltage increases. For energy scavenging, the nodes use solar cells. The solar cells use the environment, more specifically the sun, to charge the energy storage components. These design techniques have enabled the development of very small nodes that can operate for years by extracting energy from the environment.

Table 2.1 shows the evolution of sensor nodes as presented by Chong *et al.* [13]. It can be seen from table 2.1 that the state of the art technology available today is already far beyond that which was available in 2003 and very close to what was predicted for 2010. The sensor nodes developed by TUV already employ solar panels and theoretically achieve lifetimes of some years.

Table 2.1: The evolution of sensor nodes [13].

	1980's – 1990's	**2000 - 2003**	**2010**
Manufacturer	Custom contractors	Commercial: Crossbow Technology Inc., Sensoria Corp., Ember Corp	Dust Inc. and others
Size	Large shoe box and up	Pack of cards to small shoe box	Dust particle
Weight	Kilograms and up	Grams	Negligible
Node Architecture	Separate sensing, processing and communication	Integrated sensing, processing and communication	Integrated sensing, processing and communication
Topology	Point-to-point, star	Client server, peer to peer	Peer to peer
Power Supply Lifetime	Large batteries; hours, days and longer	AA batteries; days to weeks	Solar; months to years
Deployment	Vehicle-placed or air-dropped single sensors	Hand-placed	Embedded, "sprinkled", left behind

When comparing the MIT nodes from the mid 1980's to the recent TUV WSSN nodes, it can be seen that there have been tremendous advances in technology in the twenty year period. The WSSN nodes can last for years on limited energy resources, are very small, have integrated sensing, processing and communications, can use a variety of sensors for a variety of applications, can be deployed easily and in large numbers and are low-cost. The MIT nodes in turn needed a generator for energy, were very big, had separate sensing, processing and communication systems, were specifically designed for acoustic sensing, were not easily deployed, were deployed in small numbers and were expensive. If the size of a self-sustaining sensor node could be decreased from the size of a motor vehicle, in the case of the MIT node, to the size of a one Euro coin, in the case of a WSSN node, in a matter of twenty years, how small will these nodes be in twenty years from now? A self-sustaining sensor may be as small as five square millimetres within ten years (Figure 2.4) and even smaller in twenty years time.

Figure 2.4: WSN node in the near future [16].

2.2 Differences between WSNs and Other Wireless Ad Hoc Networks

A wireless ad hoc network (WANET) is a temporary network that is set up between peer nodes to satisfy an immediate need [17]. Many protocols exist for wireless ad hoc networks, but are unsuitable for WSNs due to the unique requirements of WSNs. According to Akyildiz *et al.* [18], WSNs differ from other WANETs in seven areas, namely: *network size*, *node density*, *node proneness to failure*, *frequency of topology changes*, *communication paradigm employed*, *resource limitations of nodes* and *node identification*. Each of these areas is discussed in the following paragraphs.

The *network size* of a WSN can be anything from a few nodes up to many thousands of nodes. Other WANETs on the other hand usually consist of less than a hundred nodes. A Bluetooth piconet, which can consist of up to a maximum of eight nodes, is an example of a WANET. A wireless local area network (WLAN) is another example of a WANET. WLAN is based on the IEEE 802.11b standard, which was developed by the Institute of Electrical and Electronic Engineers (IEEE). The size of a WLAN is limited to 32 nodes per access point [19].

Node density in a WSN is usually high, with a large number of nodes in a relatively small area, while other WANETs mostly consist of only a few nodes in close proximity of each other. This is due to the size of nodes. A WSN node can be as small as a one Euro coin, while nodes of other WANETs are mostly notebook computers, palmtops or cellular telephones.

A WSN might be deployed in a remote or inaccessible area, such as a jungle or a disaster area. In such circumstances the *node proneness to failure* is high due to the possibility of nodes being damaged and failing. Some nodes might also drain their energy resources quicker than other nodes due to being on a routing path that is utilized more than other paths. Nodes in other WANETs have rechargeable energy supplies and are not subjected to adverse environmental conditions that could damage them to the extent of not being able to function any longer.

The *frequency of topology changes* in a WSN is high, due to factors such as node failures, node additions, nodes moving and environmental interference. The network has to be able to adapt to these changes in node position and number. Topology changes can happen as frequently as every few milliseconds. In other WANETs, nodes usually request to join the network and leave the network after a certain period of time, which is rarely less than a couple of minutes.

The *communication paradigm employed* in WSNs includes a large number of broadcasts that are sent through the network. These broadcasts are used for network set up and maintenance, discovery of neighbours and sending of data. Other WANETs usually use point to point communications, since the source knows how to reach the destination.

The *resource limitations of nodes* in WSNs include limited energy and bandwidth resources, compared to other WANETs. The energy resources of WSN nodes cannot be replenished, while other WANETs' nodes have rechargeable batteries. The limited data rate of up to a few kilobits per second in WSNs is small compared to data rates of between one and a few hundred megabits per second in other WANETs. The memory of WSN nodes is limited to a few kilobytes, while other WANETs' nodes can have gigabytes of memory. The processors employed in WSN nodes are limited. The TUV WSSN nodes, for example, use 4MHz processors. This is very limited, compared to the GHz processors of notebook computers.

Node identification by means of globally unique identifiers are not always possible in WSNs, due to the possibly very large number of nodes in the network and the overhead caused by having a unique identifier for each node. In other WANETs, the nodes have unique identifiers such as internet protocol (IP) addresses.

The WSN is a new and unique class of WANET that differs considerably from other WANETs. The unique nature of WSNs implies that protocols designed for other WANETs cannot be implemented in WSNs and, therefore, new protocols have to be developed.

2.3 Factors Influencing WSN Design

Many different types of sensors can be employed in a WSN, including: temperature, vibration, infrared and acoustic. A WSN incorporating these sensors can then be used to monitor ambient environmental conditions such as temperature, humidity, movement, pressure and sound. The area in which these sensor nodes are scattered is known as a *sensor field* [18].

Even though WSN applications differ greatly, there are some common design factors that influence all WSNs. They include *reliability, scalability, production costs, hardware constraints, network topology, operating environment, transmission media* and *energy consumption* [18]. It is important to consider these factors when designing a protocol for WSNs and therefore, each factor is discussed individually in the following sections.

2.3.1 Reliability

Environmental interference, physical damage or a depleted energy source may cause a sensor node to fail. It is, however, important that a single node failure should not affect the overall network performance. Reliability in a WSN is the ability of the network to sustain its functionality regardless of the failure of nodes. Hoblos *et al.* [20] modelled the reliability $R_k(t)$ of a sensor node using the Poisson distribution. The probability of a node not having a failure within the interval $(0, t)$ is given by:

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

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

2.3.2 Scalability

A WSN may consist of hundreds to thousands and eventually even millions of nodes. WSN protocols have to be designed to be able to work with these large numbers of nodes and also utilise the high density of nodes. The density of a WSN can be anything from a few nodes to a few hundred nodes per square metre. According to [21], the density μ can be calculated by:

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

where N is the number of nodes in region A and the transmission range of the employed radio is R. The result of equation 2.2 is the number of nodes within transmission range of a specific node.

2.3.3 Production Costs

The production cost of a single sensor node becomes a very important issue in WSNs due to the large number of nodes in the network. For a WSN of thousands or millions of nodes to be financially feasible, the cost of a single node has to be much less than US$1 [18].

2.3.4 Hardware Constraints

There are four basic components that can be found in all sensor nodes. These components are: *a power unit, a processing unit, a sensing unit* and *a transceiver*. Some sensor nodes also contain optional components such as a *location finding system, a mobilizer* or *a power generator*. Figure 2.5 shows the basic components of a sensor node.

The power unit is very important in a sensor node. It is responsible for providing all of the other units with energy so that the node can perform its functions. A power generator or power scavenging unit can support the power unit. Solar cells could be used as power scavenging units.

The processing unit consists of a processor and some storage or memory. This unit is responsible for managing the tasks of the sensor unit. The sensing unit generally consist of a sensor and an analogue to digital converter (ADC). The ADC converts the analogue data from the sensor to digital data that can be processed by the processor. The transceiver connects the sensor node to the network. The transceiver can use either radio frequency (RF) or optical communications, such as infrared, to wirelessly connect to the network.

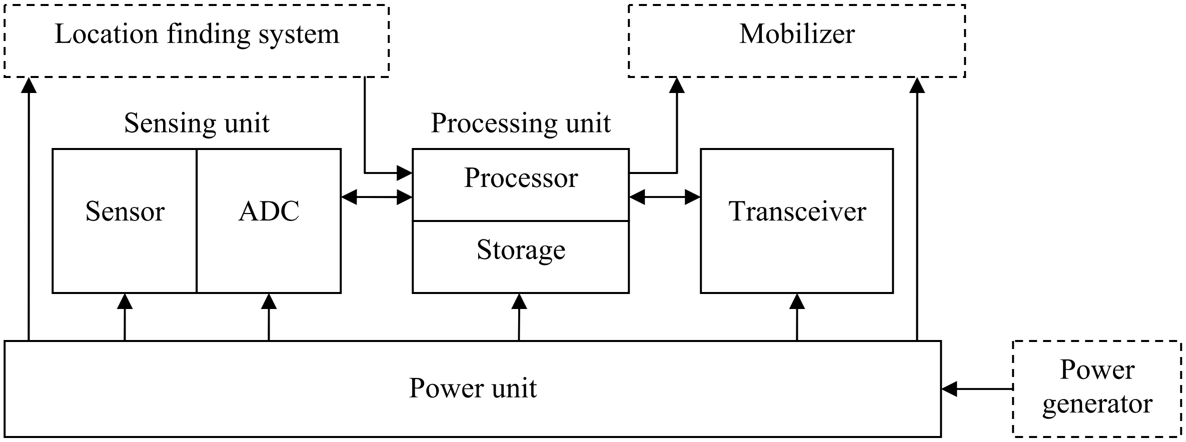

Figure 2.5: The components of a sensor node [18].

The optional location finding system might consist of a low-power global positioning system (GPS) unit. The location finding system could also be a coordinate establishment algorithm that is implemented in the processing unit to find the unit's relative position to the sink. A mobilizer unit may be used to enable the node to move, if mobility is required for the node to accomplish its task. All of these components must be able to fit in a module that is smaller than a matchbox.

2.3.5 Network Topology

Topology changes and maintenance can be viewed in three phases, namely *deployment phase, post-deployment* and *re-deployment* [18]. The initial topology is set up during deployment. The nodes can be deployed by placing them one by one, or they can be deployed in masses by, for example, dropping them from an aeroplane. Topology changes during the post-deployment phase are due to node failures and positional changes of nodes due to mobility. During the re-deployment phase, additional nodes are deployed in the network. This can happen at any time.

2.3.6 Operating Environment

The environment in which WSNs operate can range from very hospitable to extremely hostile. They can be deployed in the home, in a factory, in machinery, on the battlefield, on the bottom of the ocean, in disaster areas, in toxic areas and possibly in every other imaginable area. This requires nodes, and the network itself, to be designed with extreme environments in mind.

2.3.7 Transmission Media

Nodes in a WSN communicate with each other through the use of wireless transmission media. RF, infrared or other optical media may be used. It is important that the type of media used is available globally so that usage is not confined to certain areas. In this respect infrared and the Industrial Scientific and Medical (ISM) band for RF communication are good examples. The ISM band is globally available, unlicensed and is centred around 2.4GHz. RF communication is used by the sensor nodes developed by TUV for the WSSN project and by the sensor nodes developed by the University of California, Los Angeles (UCLA) for the Wireless Integrated Network Sensors (WINS) [22] project. The Smart Dust project of the University of California, Berkeley investigates the possibility of using specialised optical communication between nodes [23]. Two communication schemes are examined. One is passive and uses a corner cube retroreflector, while the other is active and uses a laser diode and sub-millimetre MEMS mirrors to direct the beam.

2.3.8 Energy Consumption

A sensor node is equipped with a limited energy source and hence has a lifetime that is dependent on that source. In a WSN, each node can originate data and also has to route data. When a few nodes deplete their energy resources, topology changes occur which may require rerouting of data packets. A sensor node’s task is to sense data, perform some processing and then transmit the data. Energy consumption in a node can therefore be divided into three areas: *communication, processing* and *sensing*.

A node traditionally expends most of its energy during communication. The transceiver unit consumes energy during both start-up and active states. Sensor nodes use short data packets for communication. As the size of these packets become smaller, the significance of power consumed during transceiver start-up increases. Shih *et al.* [24] calculated the power consumption (P_C) for communication as:

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

where N_T is the number of times the transmitter is switched on per unit time, 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 that the receiver is turned on per unit time, P_R is the power consumed by the receiver, R_{on} is the receiver on time and R_{st} is the receiver start-up time.

The energy consumed during processing is small and can be kept to a minimum by simplifying the processing task as much as possible. The energy consumed for sensing depends on the nature of the sensing task. Sensing can be continuous or at discreet intervals and different types of sensors have different complexities and power consumption.

2.4 Wireless Networking Standards

In March 1999, the IEEE established the 802.15 working group as part of the IEEE Computer Society's 802 Local and Metropolitan Area Network Standards Committee. The 802.15 working group was established with the specific purpose of developing standards for short distance wireless networks, otherwise known as wireless personal area networks (WPANs).

There exist four task groups within the 802.15 working group. Task group one (802.15.1) defined a standard for WPANs based on the physical (PHY) and MAC layers of the Bluetooth specification version 1.1 [25]. Task group two (802.15.2) is developing a model for the coexistence of WLAN (802.11) and WPAN (802.15). The goal of task group three (802.15.3) is to develop standards for high data rate WPANs (20Mbps and greater). Task group four (802.15.4) is responsible for developing PHY and MAC layer standards for low data rate, low complexity solutions that will achieve battery lifetime of months to years.

In the following sections we will look at the 802.15.1 and 802.15.4 standards and at the IEEE P1451.5 project, which will establish a standard for the data format and communication methods of sensors and actuators. The 802.15.4 standard is especially important since it is aimed specifically at sensors and other devices needing long battery lifetimes.

2.4.1 IEEE 802.15.1 and Bluetooth

Bluetooth was designed to be a short range (less than 10m), low cost (less than $5), and low power (1 to 100mW) wireless cable replacement technology to provide communication between portable devices and desktop machines and peripherals. The Bluetooth radio transceivers operate in the globally available, unlicensed 2.4 GHz ISM radio band. Bluetooth uses frequency-hopping spread spectrum (FHSS) and hops at a rate of 1600 hops/s [25].

The basic unit of a Bluetooth network is a *piconet*. A piconet consists of between two and eight nodes. One node is the master and up to seven active slave nodes may be connected to it. The limit of seven slaves is due to the three bit address used for active slaves in a piconet. Three bits allows eight addresses but the all zeros address is reserved for broadcasts, thus seven addresses are available for slaves. The master's clock is used to synchronize communication within a piconet. All communication within a piconet is routed via the master. When a node participates in more than one piconet, the piconets become linked and a *scatternet* is formed. A node participating in more than one piconet is called a gateway and uses Time Division Duplex (TDD) in order to be active in only one piconet at a time.

The IEEE 802.15.1 specification defines a standard for the PHY and MAC layers of WPANs, based on the Bluetooth specification. In particular, the logical link control and adaptation protocol (L2CAP), link manager protocol (LMP), and Baseband layers of the Bluetooth protocol stack form the 802.15.1 MAC layer and the Radio layer of the Bluetooth protocol stack forms the PHY layer of 802.15.1. The MAC layer is responsible for the time synchronisation of the FHSS communication and the PHY layer specifies the communication band (2.4GHz).

2.4.2 IEEE 802.15.3a and Ultrawideband

The United States (US) military developed Ultrawideband (UWB) in the 1970s for various uses including low-power communications capable of evading mainstream eavesdropping techniques. UWB is an impulse radio as opposed to carrier-based radio which transmits data continuously [26]. UWB typically transmits signals via sub-nanosecond pulses of energy operating at about 100nW per MHz of transmission bandwidth. The IEEE's proposed UWB standard would increase the pulse duration to 4 nanoseconds.

UWB sends the various pulses of a single transmission over a relatively large part of the radio spectrum, not just at a specific frequency or narrow frequency range, as is the case with cellular-phone and other radio-based technologies. The US Federal Communications Commission (FCC) has allocated UWB the spectrum 3.1-10.6GHz, currently used by satellite based telecommunications providers. This was done to avoid potential interference with radio-based technologies that use other parts of the spectrum. UWB's data rate is typically 200-400Mbps. The proposed IEEE standards would let UWB run anywhere from 110Mbps over 10 metres to 480Mbps over 1 metre [26].

The end result of UWB is a technology that provides simplicity, very low transmit power, multipath and interference immunity, and the capability to deliver data rates in excess of 100Mbps; all the while consuming very little battery power and relatively small amounts of silicon area, translating to low cost.

2.4.3 IEEE 802.15.4 and ZigBee

In 2002 the non-profit ZigBee Alliance was formed by an association of companies. The goal of the ZigBee Alliance is to develop monitoring and control products that are reliable, low cost, low power and can be wirelessly networked using an open global standard. Task group four of the IEEE 802.15 working group started working on a standard for low data rate WPANs shortly thereafter. The IEEE and the ZigBee Alliance joined forces and decided that ZigBee would be the commercial name of the technology.

Potential applications of the 802.15.4 standard include sensors, home automation, smart badges, remote controls, and interactive toys [27], amongst others. The standard specifies two direct sequence spread spectrum (DSSS) PHY layers and the use of three license free frequency bands. One PHY layer is at 868/915MHz and uses the 868-870MHz band with one channel and the 902-928MHz band with ten channels. This PHY achieves a data rate of 20kbps in the 868-870MHz band and 40kbps in the 902-928MHz band. The other PHY is at 2.4GHz and uses the 2.4-2.4835GHz frequency band with sixteen channels. It achieves a data rate of 250kbps. Table 2.2 summarises the frequency band and data rates of the 802.15.4 standard.

Table 2.2: IEEE 802.15.4 frequency bands and data rates [27].

PHY	Band	Channel Numbering	Chip Rate	Modulation	Bit Rate
868/915MHz	868-870MHz	0	300 kchip/s	BPSK	20 kbps
	902-928MHz	1 to 10	600 kchip/s	BPSK	40 kbps
2.4GHz	2.4-2.4835GHz	11 to 26	2 Mchip/s	O-QPSK	250 kbps

The 802.15.4 standard supports two addressing modes, namely 16 bit short and 64 bit IEEE addressing. The PHY layer also has features for link quality indication, receiver energy detection and clear channel assessment. A maximum packet size of 128 bytes, with a payload of up to 104 bytes, is supported along with both contention-based and contention-free channel access. The MAC layer uses full handshaking for reliability and carrier sense multiple access with collision avoidance (CSMA-CA) for channel access.

Zigbee defines three software layers [19] (network, security and application) on top of the PHY and MAC 802.15.4 layers. The network layer supports three network topologies, namely star, mesh or peer-to-peer, and cluster tree as shown in Figure 2.6. The 802.15.4 standard specifies two types of nodes: a full function device (FFD) and a reduced function device (RFD). A FFD is able to route data, while a RFD is not. The standard also specifies that the network be coordinated by at least one FFD. A star topology promotes long battery lifetime since every RFD is connected directly to the coordinator. A mesh or peer-to-peer topology provides reliability and scalability since all the nodes are FFDs and therefore can be interconnected. This introduces multiple routing paths. The cluster tree topology combines aspects of both the star and mesh topologies and tries to provide long battery lifetime as well as reliability and scalability.

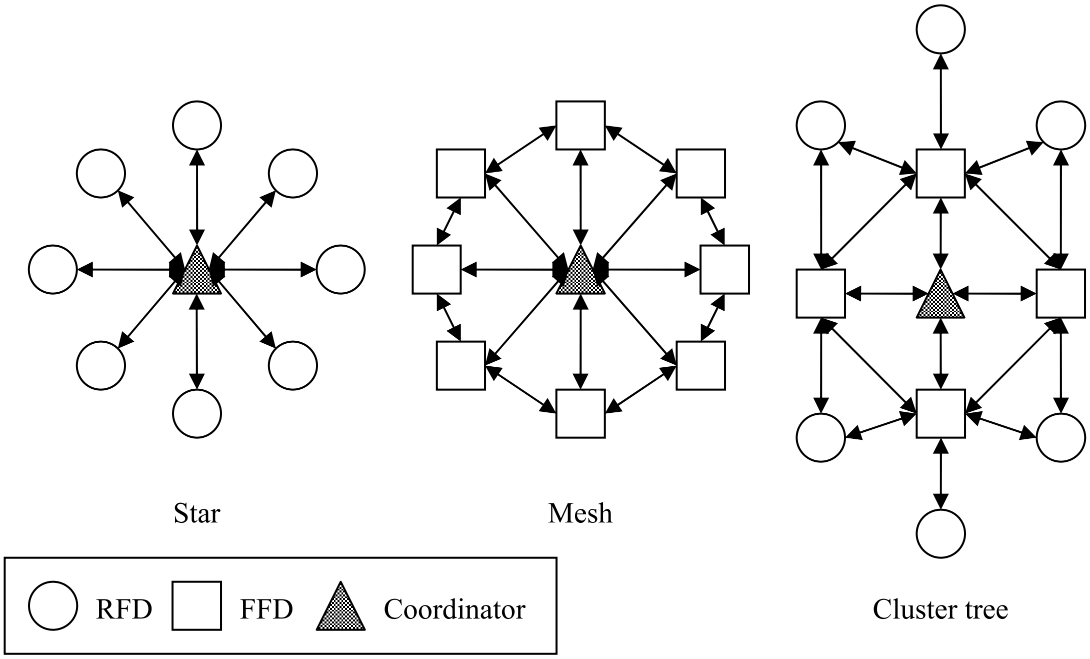

Figure 2.6: Network topologies supported by ZigBee [19].

The routing protocol used in ZigBee is based on the ad hoc on demand distance vector (AODV) routing protocol, Motorola's Cluster-Tree protocol and some ideas from Ember Corporation's GRAd. It is a hierarchical routing protocol with some table-driven optimisations. AODV is a well known reactive ad hoc network routing protocol. The AODV protocol works on a query-reply route discovery exchange, where a routing query is flooded through the network until it reaches its destination [28]. Each node along the path to the destination keeps track of which of its neighbours originated the specific route query. The destination then unicasts the reply back to the source along a path constructed during the query. This path is calculated from the next hop information stored at each node along the path.

The security layer adds the ability to encrypt the MAC layer frames with 32, 64 or 128 bit Advanced Encryption Standard (AES) encryption. The application layer defines profiles that aim to enable interoperability. It also enables nodes to determine which other devices are within their vicinity and makes it possible to match devices based on their services.

2.4.4 *The IEEE P1451.5 Project*

The IEEE P1451.5 project was initiated to develop a standard for the wireless communication methods and the data format of transducers. The goal is a smart transducer interface for sensors and actuators. Transducers use a data format known as transducer electronic data sheets (TEDS). The IEEE 1451.5 standard will define a new TEDS as well as protocols to access the TEDS and the transducer data.

Many different wireless communication interfaces and protocols are being developed for sensors by a host of different manufacturers. These interfaces and protocols are often vendor specific. An open standard that accepts various existing technologies will enhance market acceptance and enable connectivity between devices from different vendors.

2.4.5 *Comparison of Wireless Networking Standards*

There are many different standards for wireless networks. These standards divide wireless networks into categories based on factors such as network size, data rate, transmission range and battery lifetime. Table 2.3 shows a comparison of three important wireless network standards.

Table 2.3: A comparison of wireless networking standards [19].

Market Name	**Wi-Fi**	**Bluetooth**	**ZigBee**
Standard	**IEEE 802.11b**	**IEEE 802.15.1**	**IEEE 802.15.4**
Type of Network	WLAN	WPAN	WPAN
Application Focus	Web, Email, Video	Cable Replacement	Monitoring and Control
System Resources	1MB+	250KB+	4KB - 32KB
Battery Life (days)	0.5 - 5	1 - 7	100 - 1,000+
Network Size	32	7	255 / 65,000
Data rate (kbps)	11,000+	720	20 - 250
Transmission Range (meters)	1 - 100	1 - 10+	1 - 100+
Success Metrics	Speed, Flexibility	Cost, Convenience	Reliability, Power, Cost

CHAPTER 3

Wireless Sensor Network Routing Protocols

3.1 Factors Influencing WSN Routing Protocol Design

The main design challenge in WSNs is to prolong network lifetime while keeping communication integrity. This is complicated by the fact that nodes in a WSN are constrained in energy supply, processing capability and available bandwidth. There are many factors related to the inherent characteristics of WSNs that have to be considered in order to design an efficient WSN routing protocol. These factors include: *node deployment, data reporting method, node and link heterogeneity, reliability, energy consumption, scalability, network dynamics, transmission media, connectivity, coverage, data aggregation* and *quality of service* [29].

Node deployment can be either manual or random. If nodes are manually deployed, the routing in the network can be done using predetermined paths. Random deployment, on the other hand, requires routing paths to be established in an ad hoc fashion. Due to the limited transmission range of nodes, the routing path will most likely be multi-hop. The density of node deployment also plays a role in network survivability. It is preferable that the density be higher close to the sink, since all of the data from all the nodes in the network are routed through the nodes surrounding the sink.

Data reporting in WSNs depends on the time criticality of data and on the WSN application. Data reporting can be time-driven, query-driven, event-driven or some mixture of these methods. Time-driven reporting occurs when sensor nodes periodically switch on their sensors and transmitters, to transmit sensed environmental data that is of interest. Event-driven reporting occurs when nodes react to drastic changes sensed in their environment. Query-driven reporting occurs when nodes respond to queries for data. Event-driven and query-driven methods can be used for situations where data delivery is time critical. The data reporting method has a big influence on the routing protocol in terms of route calculation and energy consumption.

Node heterogeneity has to do with the fact that sensor nodes might have different capabilities or different roles within the network. This is mostly the case in hierarchical or cluster networks. Some nodes in the network are assigned the role of cluster head. Cluster head nodes might be nodes with different capabilities in terms of energy resources, processing capability and bandwidth, compared to other nodes in the network. If they have enhanced capabilities, they remain cluster heads throughout the lifetime of the network. In the case where all the nodes in the network have the same capabilities, the cluster heads are assigned at periodic intervals. These assigned cluster heads have to be rotated to prevent energy hotspots from forming. Link heterogeneity has to do with the fact that some nodes might be required to send data more frequently than other nodes in the network and that more than one data reporting method might be used in the same network. The routing protocol employed has to consider the capabilities of the different nodes in the network and has to be able to route data at different rates, if necessary.

Reliability is an important issue in sensor networks since the nodes are prone to failure (refer to section 2.3.1). Node failures may be due to nodes depleting their energy sources or physical damage or environmental interference. The overall network functionality should not be compromised when some of the nodes fail. Node failures lead to areas losing sensor coverage and might cause data to be lost if a node has received data and fails before it can transmit the data. The routing protocol has to route packets through regions of the network where there are higher levels of available energy if failures occur.

Energy consumption in sensor nodes occurs mainly due to computational processing and communication (refer to section 2.3.8). WSNs are usually multi-hop networks. If a node fails in such a network, the topology changes and rerouting of data and network reorganisation might have to occur. Therefore, it is essential to apply energy conserving techniques in computation and communication. Communication energy conservation can be achieved by limiting the packet sizes of the data to be sent and by limiting the number of packets that are routed through the network. Computational energy efficiency can be achieved by limiting the number of computational tasks to be done by a node, in other words: making the routing protocol as simple as possible.

Scalability is needed in sensor networks due to the fact that the number of nodes in the network could be anything from a few nodes to a few thousand nodes or more. A routing protocol used in a WSN has to be able to function efficiently with any number of nodes.

Network dynamics has to do with the mobility of nodes. In some networks the nodes are fixed while in others, the nodes are mobile. If sensor nodes or the sink node are mobile, routing becomes more difficult since static routes cannot be used. The events to be monitored might also be of a mobile nature, for instance if the network is used for vehicle tracking. In such a scenario the routing is reactive (routes are established as needed). Routing in a network where the events are static might use a table-driven approach (each node keeps a routing table).

Transmission media in WSNs can be any wireless communication link. WSNs also face the problems associated with wireless channels, such as fading, noise and interference. Efficient use of the transmission media is related more to the MAC layer. The MAC layer might use a time division multiple access (TDMA) channel access method or a contention based channel access method such as CSMA. The TDMA method generally consumes less energy than the CSMA method [30]. The data rate of the wireless communication media in a WSN is usually in the order of 1-100kbps, which is low compared to other wireless networks. This is an important consideration for the design of the routing protocol since large network layer packets will result in more than one packet being transmitted every time the sensor node needs to send data. The network layer packets have to be kept small in order to limit the number of required transmissions for new sensed data.

Connectivity in WSNs is assumed to be high due to the high node density. The large number of nodes in a WSN makes it unlikely that nodes will be isolated. The connectivity will however decrease as nodes fail. As the connectivity changes due to topology changes, the routing paths also change.

Coverage in WSNs deals with the range and accuracy of the sensing that a sensor node can achieve. This limited area coverage of sensor nodes makes it important to have densely deployed nodes; otherwise there might be some areas that are not covered by the WSN. If node density is low, it also causes nodes to be cut off from the network more easily since there are fewer paths to the sink. Therefore network partition occurs sooner.

Data aggregation is very important in WSNs since the density of nodes results in a great deal of redundant data being sent through the network. A node in a WSN might receive the same sensed data from more than one neighbour. Transmitting the data more than once would result in energy wastage. Data aggregation can be done using duplicate suppression, maxima, minima or average [29]. Duplicate suppression occurs when a node discards messages that it has already received. This is the simplest form of data aggregation. Data aggregation using maxima, minima and average can also be called data fusion. In the case of data fusion, the data that a node receives from its neighbours is combined and only the maximum, minimum or average of all the received messages is transmitted in a single message. Data aggregation can reduce the number of messages that are routed through the network but can introduce latency if data fusion is used, since nodes have to wait to receive data from all of their neighbours before transmitting the combined data. Timing has to be considered since a node cannot simply wait until it has received data from all of its neighbours as one neighbour might have failed. A time limit has to be set for each round of data transmissions and after that time has elapsed, the data is sent even if the node has not received data from all of its neighbours.

Quality of service in WSNs mostly deals with latency and reliability. Some WSN applications require that sensed data reach the sink within a certain amount of time in order to be useful. An example of a time critical application might be vehicle tracking. If the latency in the network is high, the vehicle might be far from the reported position by the time the data arrives at the sink. In other applications the data from nodes might not be critical in terms of latency but critical in terms of being delivered. Such critical data might have adverse effects on the network if undelivered. An example of an application with data that is critical but not time critical might be a WSN that monitors the number of working lights in a shopping centre. If a light fails, it is important that it be replaced but it does not have to be replaced within seconds.

3.2 Classification of WSN Routing Protocols

WSN routing protocols can be classified in three ways: according to the way routing paths are established [31], according to the network structure [29] and according to the initiator of communications. The classification of WSN routing protocols is shown in Figure 3.1.

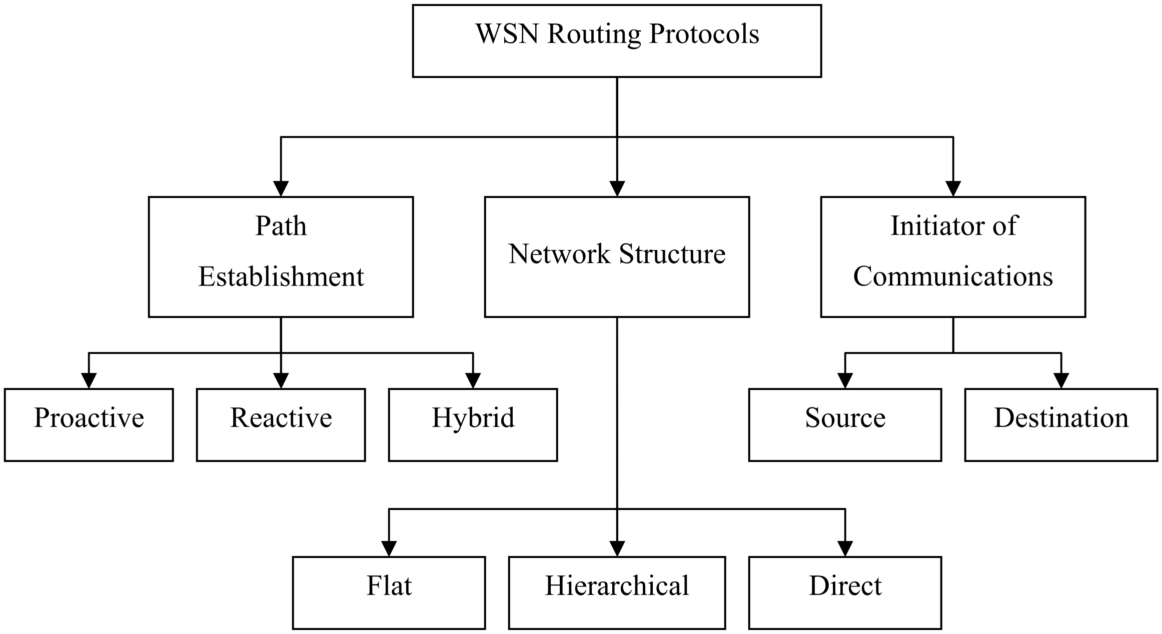

Figure 3.1: Classification of WSN routing protocols.

Routing paths can be established in one of three ways, namely proactive, reactive or hybrid. Proactive protocols compute all the routes before they are really needed and then store these routes in a routing table in each node. When a route changes, the change has to be propagated throughout the network. Since a WSN could consist of thousands of nodes, the routing table that each node would have to keep could be huge and therefore proactive protocols are not suited to WSNs. Reactive protocols compute routes only when they are needed. Hybrid protocols use a combination of these two ideas.

There are basically three possible network structures for WSNs. They are: flat, hierarchical and direct. Direct communication is impractical in WSNs since it requires that all of the nodes be one hop away from the sink. In flat protocols, all the nodes in the network are equal and can participate equally in the routing task. Nodes close to the sink participate more than nodes further away since all of the messages destined for the sink pass through the nodes around it. In hierarchical or clustering protocols, the network is subdivided into clusters of nodes and each cluster has a cluster head. The nodes within a cluster send messages only to the cluster head. The cluster head then in turn forwards all messages of its cluster towards the sink.

Communication within a WSN can be initiated either by the source of data or by the destination of the data. In source initiated protocols, nodes send data to the sink when they have data of interest. Source initiated protocols use time-driven or event-driven data reporting. This means that data is sent either at certain intervals or when nodes sense certain events. Destination originated protocols use query-driven reporting and nodes respond with data to queries that are sent by the sink or another node. Destination-initiated protocols incur a large amount of overhead since requests are usually flooded through the network. This means that every request for data will result in a flooding of the network.

Most routing protocols can be divided into either flat or hierarchical protocols at the highest level and can then be further divided into source initiated or destination initiated protocols. Routing path establishment in WSNs is generally reactive.

A discussion of some of the available routing protocols is given in the following sections. Flat routing protocols are discussed first and then hierarchical routing protocols. At the end of each of these two sections is a discussion of the advantages and disadvantages of the specific network structure as related to WSNs. The last section of this chapter gives a comparison of the discussed routing protocols.

3.3 Flat Routing Protocols

3.3.1 Flooding and Gossiping

Flooding and gossiping [32] were two of the first routing protocols to be applied to WSNs. The disadvantages that flooding has for WSNs include *implosion*, which occurs when duplicate messages are sent to the same node, *overlap*, which occurs when two nodes that are in the same region send similar messages to the same neighbour and *resource blindness*, which is caused by the nodes not taking energy constraints into consideration [29]. Gossiping avoids implosion by randomly selecting one or a subset of neighbours and then sending the message only to those neighbours. This does, however, introduce propagation delay. These two protocols were not specifically designed for energy constrained networks and therefore do not provide energy efficiency.

3.3.2 Sensor Protocols for Information via Negotiation

Sensor protocols for information via negotiation (SPIN) [4] was one of the first routing protocols to be proposed for WSNs. SPIN is a source initiated protocol that uses a flat network structure and reactive routing. SPIN assumes that nodes in close proximity posses the same data and hence nodes only transmit data that its neighbours do not already have. The data available at each node is distributed through the whole network. Therefore every node in the network has the data of every other node in the network. An aspect of SPIN that improves WSN routing is the fact that the sensor nodes only maintain routing information about their direct neighbours. This makes the protocol scalable.

Nodes use meta-data to describe the data that they have available. This meta-data is orders of magnitude smaller than the actual data in terms of the number of bits used. Meta-data is used to eliminate sending redundant data through the network. SPIN does not specify the format of the meta-data as this is assumed to be application specific. The main idea behind the protocol is to address the deficiencies of flooding by eliminating the possibility of sending duplicate data. SPIN uses time-driven reporting. Data is sent through the network using an approach of advertisement and request that works in three stages. In the first stage a node having data of interest sends an advertisement message (ADV) containing meta-data to its neighbours. During the second stage, neighbours interested in the advertised data respond with a request (REQ) for the data. The final stage is the stage where the originating node sends the data

(DATA) to all of its neighbours that requested the data. This process is then repeated by all of the nodes that have received the data. Figure 3.2 shows the three stages of the SPIN data negotiation process.

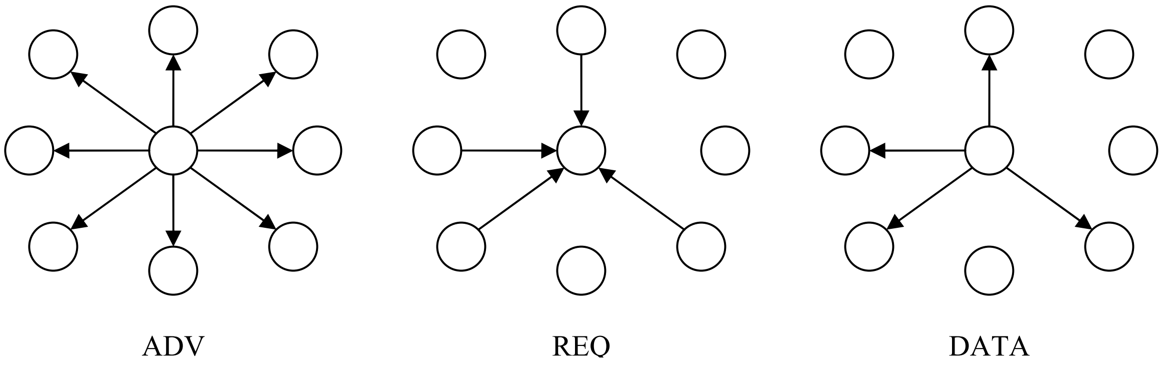

Figure 3.2: The three stages of SPIN.

SPIN produces an incredible number of messages if all of the data produced at each node in the network is requested by each of its neighbours. The fact that data reporting is time-driven instead of event-driven causes nodes to send advertisements even if no new data is available. Another problem with SPIN is the fact that if some nodes between the source and the sink do not request advertised data, the data never reaches the sink. This protocol only works if the network is small and only a few neighbours request available data when data is advertised.

3.3.3 Directed Diffusion

Directed diffusion [5] is a popular routing protocol and is probably the best known WSN routing protocol. Many other routing protocols have been based on it. Directed diffusion is a destination initiated protocol that uses a flat network structure and reactive routing. The protocol uses data-centric routing, where queries are directed at certain areas in the network and not the whole network. Directed diffusion consists of three stages, namely interest diffusion, gradient setup and data delivery.

During interest propagation, the sink node floods an interest for named data through the network. Named data consists of attribute and value pairs. An attribute might be “temperature” and the value at one node might be “30”. A request for named data might be for all nodes that have the attribute "temperature" and values ">30" to respond with their data. The idea of using named data for requests is an efficient way to eliminate the possibility of receiving undesired or irrelevant data. The initial interest also specifies the initial rate at which the nodes have to send data to the sink, which might be every ten seconds, and includes a timestamp that specifies when nodes can stop sending data, for example after ten minutes. Nodes add the interest to an interest cache. The interest cache contains an entry for each received interest. The interest entry contains the ID of each neighbour from which the interest was received and the data rate towards that neighbour.

During the second stage of directed diffusion, nodes having attribute-value pairs matching the interest start sending data to all of the neighbours in the interest cache according to the specified data rate. Gradients are also set up for the interest. A gradient is simply the data rate at which to send data about a specific interest to a specific neighbour. Directed diffusion also incorporates data aggregation. Nodes receiving data directed at the sink add the data to a data cache. Nodes will check the data cache each time a data message is received to see if the data is new. If the data has already been seen the node will disregard the message. When data reaches the sink, it reinforces one or more paths by sending another interest. This interest is for the same named data but it is sent to a specific destination node along one path and specifies a higher data rate and a longer time before transmission should be stopped. This path might be calculated by sending data only to the node from which the interest response was first received at each hop.

During the last stage of the protocol, a node that has been reinforced sends data towards the sink at the data rate specified in the reinforcement message. The data is sent along the single path that was established. Figure 3.3 shows the three stages of directed diffusion.

One problem with directed diffusion is the overhead associated with the protocol. Interests are flooded, which consumes excessive amounts of energy. The initial replies from nodes are also flooded, adding to the energy waste. Another problem is the possibly large memory requirements for the nodes. Each node stores a table containing all the interests that it has received. Each interest entry also has one sub-entry for each node from which it received that interest. In a large sensor network of thousands of nodes there might be many interests and interest tables can grow exponentially. Therefore, directed diffusion cannot be used for applications where data is needed from all of the nodes at frequent intervals.

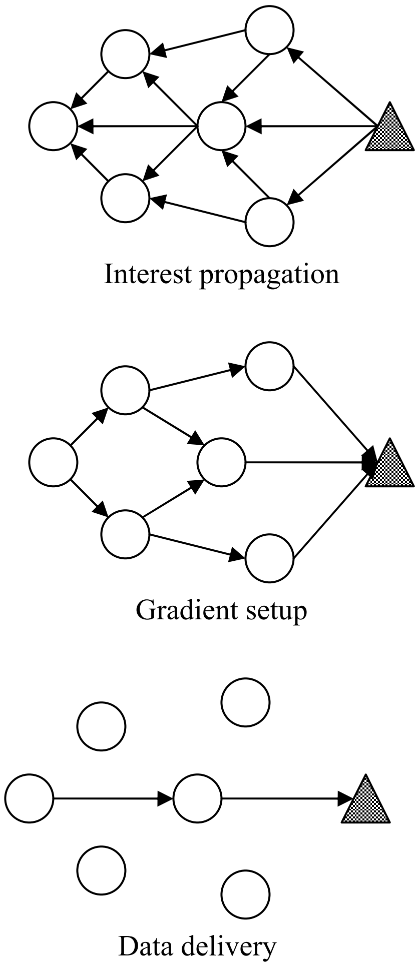

Figure 3.3: The three stages of directed diffusion.

3.3.4 Rumor Routing

Rumor routing [6] is derived from directed diffusion. It is a destination initiated protocol that uses a flat network structure and hybrid routing. The goal of rumor routing is to flood the events rather than the interest. The idea behind this approach is that the interest can be routed to nodes that have observed the event rather than flooding the interest through the entire network. To flood an event, rumour routing uses packets that have a certain time to live (TTL), called agents. If a node senses a certain event it adds the event to its event table and transmits an agent with a certain TTL. An agent contains a table of the events as observed by the node. As the agent travels through the network, all of the nodes receiving the agent update their event tables. If an agent observes another event, it updates its event table and propagates the new event along with the original event. An agent chooses the next hop to where it will be transmitted by keeping a list of all the nodes it has visited and, when it arrives at a new node, chooses a neighbour that is not in the list. When an agent arrives at a new node its TTL is decremented before it is sent further. If the TTL is zero, the agent is discarded and not sent further. When the sink sends an interest, it sends the interest on a random walk until it finds a node with a path to the required event.

The protocol is intended for networks with few events and many interests and will not provide energy efficiency otherwise. The size of the agent packet might grow very large due to the events table and the list of visited nodes, both of which might grow quite large. An agent contains only one route to each event and therefore if many queries are generated for the event, nodes along that path will deplete their energy sources quickly. The way in which agents select their next hops also influences the network greatly since queries will be routed along that path.

3.3.5 Minimum Cost Forwarding Algorithm

The minimum cost forwarding algorithm (MCFA) [7] assumes that the direction of routing is always known. MCFA is a source initiated protocol that uses a flat network structure and proactive routing. Nodes do not need to have unique IDs and instead of maintaining routing tables, nodes maintain the least cost estimate from them to the sink. The sink node sets its minimum cost to zero while all the nodes in the network set their initial costs to infinity. The sink then broadcasts a message with its cost. A node receiving the broadcast updates its cost if the cost in the message plus the cost of the link is lower than the current cost. The link cost could be hop count, delay or some other metric. If the new cost is lower, the node rebroadcasts the message, otherwise it discards it. This establishes a minimum cost at each node in the network. When a message is to be sent, a node broadcasts it to all of its neighbours. A node that receives such a message checks whether it is on the least cost path to the sink. If it is on the least cost path, it rebroadcasts the message to all of its neighbours. This is repeated until the message reaches the sink. One problem of MCFA is that nodes will deplete energy along certain paths if the minimum cost is not updated regularly. Another problem is that if hop count is used as the cost or if nodes are uniformly distributed and energy expenditure is used as the cost, multiple nodes will consider themselves on the minimum cost path and the protocol is reduced to flooding.

3.3.6 Gradient-Based Routing

Gradient-based routing (GBR) [8] is another protocol based on directed diffusion. GBR is a destination initiated protocol that uses a flat network structure and reactive routing. GBR adds hop count to the interest when it is diffused through the network. This enables each node to calculate a parameter called the height of the node, which is the minimum number of hops required to reach the sink. This height is then used as the gradient. If there are multiple neighbours with the same height, the protocol can use one of three methods to choose one of them. The first method is based on choosing one of the neighbours at random. The second method is for nodes to increase their height if their energy drops below a threshold. The third method is for nodes not to route new message streams through neighbours that are already on a different message stream. GBR faces the same flooding overhead problem that directed diffusion faces, since interests are still flooded through the network.

3.3.7 Energy Aware Routing

Energy aware routing [9] is similar to directed diffusion but maintains multiple paths at each node instead of just one. It is a destination initiated protocol that uses a flat network structure and proactive routing. Multiple paths are found between source and destinations, and each path is assigned a probability of being chosen, depending on the energy metric. Every time data is to be sent, one of the paths is randomly chosen depending on the probabilities. Diffusion sends data along all paths at regular intervals, while energy aware routing sends data only along one path at all times. The protocol has three phases. The first phase is the setup phase. During this phase localized flooding is used to build routing tables. The second phase is the data communication phase. During this phase data is sent from the source to the destination. The final phase is the route maintenance phase which uses periodic localized flooding to maintain routes. One of the main problems with this protocol is the assumptions that have to be made for the protocol to work. Assumptions such as that all the nodes are location aware and that interests are sent only to nodes that are closer to the source and further away from the destination, in the direction of the source. Another problem is the fact that the protocol requires that nodes have two transceivers. The route setup is also very complicated due to the fact that nodes have to be addressed according to location and the node type.

3.3.8 Routing Protocols with Random Walks

Routing based on random walks [11] uses multi-path routing to try to achieve load balancing. It is a source initiated protocol that uses a flat network structure and reactive routing. This routing protocol is designed specifically for large networks with static nodes. Each node in the network has a unique ID, but the topology of the network is not practical. Each node in the network is arranged such that it falls on exactly one crossing point of a grid on a plane. It uses a distributed asynchronous version of the Bellman-Ford algorithm to calculate the distance between nodes when finding a route. The Bellman-Ford algorithm is a general case of Dijkstra's algorithm. Dijkstra's algorithm is used to calculate single-source shortest paths in a weighted graph, where the weights of the edges of the graph are positive. The Bellman-Ford algorithm does exactly the same except that it is able to work with negative weights. Intermediate nodes in a random walk network will choose the neighbour that is closest to the destination according to a certain probability. Load balancing is achieved by manipulating this probability. The main problem with this protocol is the required network topology, which is impractical.

3.3.9 Minimum Energy Communication Network

Minimum energy communication network (MECN), proposed by Rodoplu *et al.* [10] is a location-based protocol that tries to achieve energy efficiency by minimizing the transmission power used by nodes. It is a source initiated protocol that uses a flat network structure and reactive routing. The protocol creates a relay region for nodes. A relay region for a node consists of other nodes through which the node can transmit data towards the sink in an energy saving manner. The transmission power needed to reach a node at distance d in any wireless network using radio communication, is proportional to d^{α}, where α is the path loss factor generally accepted to be $(2 \leq \alpha \leq 4)$ [3]. Taking this fact into consideration, the protocol chooses a path from the source to the sink that has more hops with shorter transmission distances, assuming that nodes have variable transmission power. The protocol also assumes that every node in the network is within transmission distance of every other node in the network. The nodes in the network need to be location aware and are assumed to have GPS receivers. This requirement causes the nodes to use excessive amounts of energy. The requirement that every node is within communication distance of every other node is also not practical for WSNs. Small minimum energy communication network (SMECN) [12] was derived from MECN and has all of the problems that MECN has. The only differences are that SMECN considers the possibility that there might be obstacles between two nodes, hampering communication and the relay region is smaller.

3.3.10 Advantages of Flat Routing Protocols

Scalability: Flat networks are scalable due to the fact that each node participates equally in the routing task and since nodes only need information about their direct neighbours.

Simplicity: Flat networks make it possible for routing protocols to be simple, since it is possible to establish the network without much overhead and there is no need for complex cluster head selection algorithms.

3.3.11 Disadvantages of Flat Routing Protocols

Hotspots: If nodes are uniformly distributed throughout the network and there is only one sink node, the nodes around the sink will deplete their energy sources sooner than other nodes. This is due to the fact that all of the network traffic is routed through the nodes surrounding the sink.

3.4 Hierarchical Routing Protocols

Hierarchical routing protocols are generally concerned with the establishment of cluster heads and the way in which nodes decide which clusters to join. Almost all of the hierarchical routing protocols are concerned with one or both of these issues. Routing path establishment is usually not considered since nodes are one hop from the cluster head in most cases and always send data to the cluster head. The fact that hierarchical protocols mainly describe different approaches for cluster head selection or the way a node selects which cluster to join makes it unnecessary to describe each protocol. Therefore only a few hierarchical protocols will be discussed and the problems that all hierarchical protocols face will then be examined.

3.4.1 Low Energy Adaptive Clustering Hierarchy

Low energy adaptive clustering hierarchy (LEACH) [33] was one of the first hierarchical protocols for WSNs. LEACH consists of two phases. The first phase is the setup phase during which clusters are organized and cluster heads are selected. During the second phase, the steady state phase, data is sent to the sink. During the setup phase a certain fraction of nodes elect themselves as cluster heads. This election process is done as follows. Every node in the network chooses a random number between zero and one. If the chosen number is below a certain threshold, the node becomes a cluster head. All the nodes that have been elected as cluster heads broadcast an advertisement through the network indicating that they are cluster heads. The other nodes in the network then decide which cluster they want to join based on the received signal strength of the advertisements. The nodes then inform the cluster head whose cluster they want to join that they want to be part of its cluster. The cluster head assigns all of the nodes that join its cluster a specific timeslot in which they can transmit, and then broadcasts the schedule to all the nodes in the cluster. When nodes have been assigned their timeslots, the steady state phase begins and nodes start transmitting data. After a certain predetermined time the network goes back into the setup phase and new cluster heads are elected in order to distribute energy consumption.

LEACH has many problems. One is the fact that all of the nodes are required to be able to communicate directly with the sink. It is also required that the nodes be able to support different MAC protocols, since the cluster heads use two MAC protocols. The cluster head selection is also not efficient since nodes electing to be cluster heads might be concentrated in one part of the network. Moreover, the message overhead is large, as in most hierarchical protocols.

3.4.2 Hierarchy-based Anycast Routing

Hierarchy-based anycast routing (HAR) [34] builds a routing tree that is rooted at the sink. The sink node initiates the building of the routing tree by broadcasting a packet that requests child nodes. Nodes receiving the packet wait for a certain amount of time to see whether other possible parent nodes send child request packets. All of the parent advertisements are stored in a table in each node. A node then chooses the best parent from its parent table and sends a request to that node to become a child. The parent then replies with an accept packet to each child node. If a child node does not receive an accept packet within a certain period of time, it rebroadcasts the request to join. This rebroadcast can occur up to two times before the node chooses another parent. When a node has received an accept packet, it repeats the process by sending a request for children. Each node only knows about its parent and grandparent and messages from a specific node will always be sent to the same parent until the parent fails. This causes hotspots in the network. When a node fails, the tree is setup again using the above process. The biggest problem with the protocol is the very large number of messages that are sent during the setup of the routing tree.

3.4.3 Hierarchical Energy-Aware Routing for Sensor Networks

Hierarchical energy-aware routing for sensor networks (HEAR-SN) [35] aims to achieve energy efficiency through data aggregation. The authors acknowledged that *data-centric* routing protocols are not applicable to networks where data is required from all or most of the nodes in the network. Cluster heads in HEAR-SN are nodes with more energy resources and better radio equipment than other nodes in the network. Cluster heads periodically send out a beacon. Nodes receiving the beacon evaluate the link information and decide whether or not to update their routing tables. An update occurs if the link information is better than the previously stored link information. If an update is made, the node retransmits the beacon. The beacon is only sent to nodes within the specific cluster, which means that nodes always remain within the same cluster and always have the same cluster head. The beacon packet is used by nodes to find the best route to their cluster head. The main problem with HEAR-SN, as with many hierarchical protocols, is the fact that the cluster head nodes have to have better radio equipment and more energy resources than the other nodes in the network.

3.4.4 Balanced Aggregation Tree Routing

The balanced aggregation tree routing (BATR) [36] protocol establishes a routing tree based at the sink node. The protocol assumes that the sink node knows the exact location of every node in the network. The sink node computes the routing tree and then sends each node in the network information specific to the node. The information includes the node's parent as well as the timeslot in which the node can transmit. After a certain number of rounds, the sink recalculates the tree, taking failed nodes into account, and sends the new routing information to each node. The main problem of BATR is that nodes on certain paths deplete their energy quicker than other nodes in the network.

3.4.5 Advantages of Hierarchical Routing Protocols

Data aggregation: The advantage of hierarchical routing is that data from the entire cluster can be combined by the cluster head and then sent towards the sink.

3.4.6 Disadvantages of Hierarchical Routing Protocols

Hotspots: Nodes elected as cluster heads consume more energy than other nodes in the network. If cluster heads are not rotated regularly, the network becomes partitioned and this causes areas to become cut off from the network.

Physical requirements: Many protocols require that the cluster head nodes have more available energy resources than other nodes in the network.

Complexity: If cluster head nodes have the same capabilities as other nodes in the network, then a technique for selecting and rotating cluster heads has to be employed in the network in order to balance energy consumption. This increases the number of energy consuming messages that are sent through the network.

Not scalable: Hierarchical protocols are not very scalable since the number of cluster heads increases as the network size increases. Therefore, the number of specialised cluster head nodes has to increase or the message overhead in cluster head establishment has to increase.

3.5 Comparison of WSN Routing Protocols

From the discussion in the previous sections it is apparent that a flat network structure is better for WSNs than a hierarchical structure. A flat network structure can be simple and is scalable. The simplicity that flat network structures can achieve, has to do with the fact that fewer messages are sent through the network and that nodes do not need to perform complex computations to elect cluster heads or to decide which cluster to join.

Flat routing protocols can however also be inefficient if they are not designed with the limitations of WSNs in mind. Table 3.1 gives a comparison of the discussed flat routing protocols.

Table 3.1: Comparison of WSN routing protocols with a flat network structure.

Protocol	Path Establishment	Initiator of Communications	Main Problem
SPIN	Reactive	Source	Large number of messages
Directed diffusion	Reactive	Destination	Large number of messages
Rumor routing	Hybrid	Destination	Large number of messages
MCFA	Proactive	Source	Regular updates needed to prevent node failures
GBR	Reactive	Destination	Large number of messages
Energy aware routing	Proactive	Destination	Computationally complex Requires nodes to have two transceivers
Random walks	Reactive	Source	Irregular network structure required
MECN	Proactive	Source	All nodes have to be within transmission distance of the sink Nodes are required to have GPS

CHAPTER 4

The Design of a Simple Energy Efficient Routing Protocol

4.1 Design Choices

As can be seen from the discussion of WSN routing protocols in Section 3.4, there exists a need for a protocol that achieves energy efficiency while still being practical. The design of a new ***Simple Energy Efficient Routing (SEER)*** protocol is discussed in this chapter. The goals of SEER are: scalability, energy efficiency, simplicity and practicality. The following sections describe how each of these goals was achieved.

4.1.1 Scalability

WSNs can consist of anything from a few nodes to a few thousand nodes. Therefore a WSN routing protocol has to be scalable. An important factor that influences the scalability of a routing protocol is the network structure. A *flat network structure* was chosen for SEER due to the fact that it scales better than a hierarchical network structure (Section 3.4). The use of a flat network structure implies that every node in the network will be able to participate equally in the routing task.

4.1.2 Energy Efficiency

Energy efficiency in a WSN concerns prolonging the lifetime of the network as a whole, by prolonging the lifetime of each individual node. The goal is to have the energy of all of the nodes in the network decrease at more or less the same rate. If some nodes deplete their energy sources sooner than other nodes, the network might become partitioned. As discussed in Section 3.1, communication and computational processing are the two factors that consume the most of a WSN node's energy. Therefore, to ensure that nodes survive for as long as possible, SEER implements some design strategies:

1. *The protocol is source initiated*. This eliminates the need for the sink to flood an interest for data through the network and therefore reduces the number of messages that are transmitted by individual nodes.

2. *The protocol uses event-driven reporting*. Nodes only transmit data when new data is observed. New data depends on the application of the network. Sensor nodes could be placed in an environment where sensed data below a certain threshold is not of importance. New data in such an application would be data above the specified threshold.

3. *Data is routed along a single path, which is dynamically established*. Every time a node needs to send data, it selects one neighbour to send the data to. The neighbour is selected based on the neighbour's hop count and available energy.

4. *The routing protocol is computationally simple*. The method for selecting the next hop neighbour does not require complex rules or expressions to be evaluated.

4.1.3 Simplicity

The goal of simplicity has two aspects, namely *computational simplicity* and *implementation simplicity*. Computational simplicity is achieved by not requiring nodes to do any complex computations during any part of the protocol operation. Implementation simplicity is achieved by making the operation of the protocol easy to understand, so that it can be applied to nodes without difficulty.

4.1.4 Practicality

The goal of practicality sets SEER apart from most protocols in the literature. Practicality is achieved by designing the protocol such that it functions without dependencies on node capabilities or network layout. SEER does not require nodes to have specific capabilities, such as GPS positioning, dual transceivers or variable transmit power in order to function. The protocol also functions regardless of the network layout.

4.2 Protocol Operation

The steps involved in the routing of packets in a SEER network are discussed next. It is important to note that each node is required to keep a ***neighbour table*** for the protocol to function. The neighbour table contains an entry for each node within transmission distance of a specific node.

STEP 1: Network setup and neighbour discovery

Once the network has been deployed in the area where it is to operate, the sink transmits a broadcast packet. The broadcast packet contains the header fields shown in Table 4.1.

Table 4.1: Fields contained in the network layer header of broadcast messages.

Field	**Size (bits)**
Source address	16
Destination address	16
Sequence number	8
Hop count	8
Energy level	16
Total	**64**

The source and destination addresses are 16 bit addresses enabling 65536 (2^{16}) unique addresses. Each node in the network is assumed to have a unique address within the network. The 8 bit sequence number is used to identify new broadcast messages. The sink increments the sequence number every time it sends a new broadcast message. Nodes store the sequence number locally and forward broadcast messages only if the sequence number of the message is different from the stored one. The sequence number uses 8 bits in order to ensure that latency in the network does not cause nodes to mistakenly forward old broadcast messages. An 8 bit hop count ensures that nodes can be up to 255 hops from the sink.

When a node receives this initial broadcast message, it checks whether it has an entry in its neighbour table for the node that transmitted the message. If not, it adds an entry that consists of the neighbour's address, hop count and energy level. The node then increments the hop count stored in the message and stores this hop count as its own hop count. It then retransmits the broadcast, but changes the source address field to its address and the energy level field to its remaining energy level. Every node in the network retransmits the broadcast message once, to all of its neighbours. If a node receives a broadcast message with a lower hop count than the hop count it currently has, it updates its hop count. When this initial broadcast has been flooded through the network, each node knows its hop count and has the address, hop count and energy level of each of its neighbours.

STEP 2: Transmitting new data

When a node observes new data, as defined earlier, it initiates the process of routing. Two types of data packets can be sent: *normal data* and *critical data*. If a message is considered critical, for example when the sensed temperature changes from 25°C to 100°C within a very short time, a flag is set in the message indicating that it is critical. A node that originates a critical message transmits it to two neighbours instead of only one. The fields contained in the network layer header of data messages are shown in Table 4.2.

Table 4.2: Fields contained in the network layer header of data messages.

Field	**Size (bits)**
Source address	16
Destination address	16
Creator address	16
Critical flag	1
Hop count	8
Energy level	16
Total	**73**

The creator address field is used to inform the sink of which node in the network originated the data message, since the source address is changed at every hop of the routing path. It is assumed that the sink knows where each node is in the network. If the sink does not know which node originated the data and where the node is located, the data is useless.

A node bases its routing decision on two metrics, namely ***hop count*** and ***remaining energy***. A node searches its neighbour table for all its neighbours with smaller hop counts than itself. If there is only one such neighbour, that neighbour is selected as the destination for the message. If there is more than one neighbour with a smaller hop count, the node selects the neighbour who has the highest remaining energy entry in the neighbour table.

If a node does not have a neighbour with a smaller hop count, it searches for a neighbour with a hop count that is the same as its own. If there is only one such neighbour, that neighbour is selected. If more than one neighbour has the same hop count, the neighbour with the highest remaining energy is selected. If a node does not have any neighbours with hop counts smaller or equal to its own hop count, the message is discarded.

Before the message is sent, the remaining energy entry for the selected neighbour is decreased in the neighbour table. If the message is a critical message, the process of selecting a neighbour is repeated and the message is sent to a second neighbour. Using hop count as the routing metric ensures that the message is always sent in the direction of the sink. The process of neighbour selection is depicted in Figure 4.1.

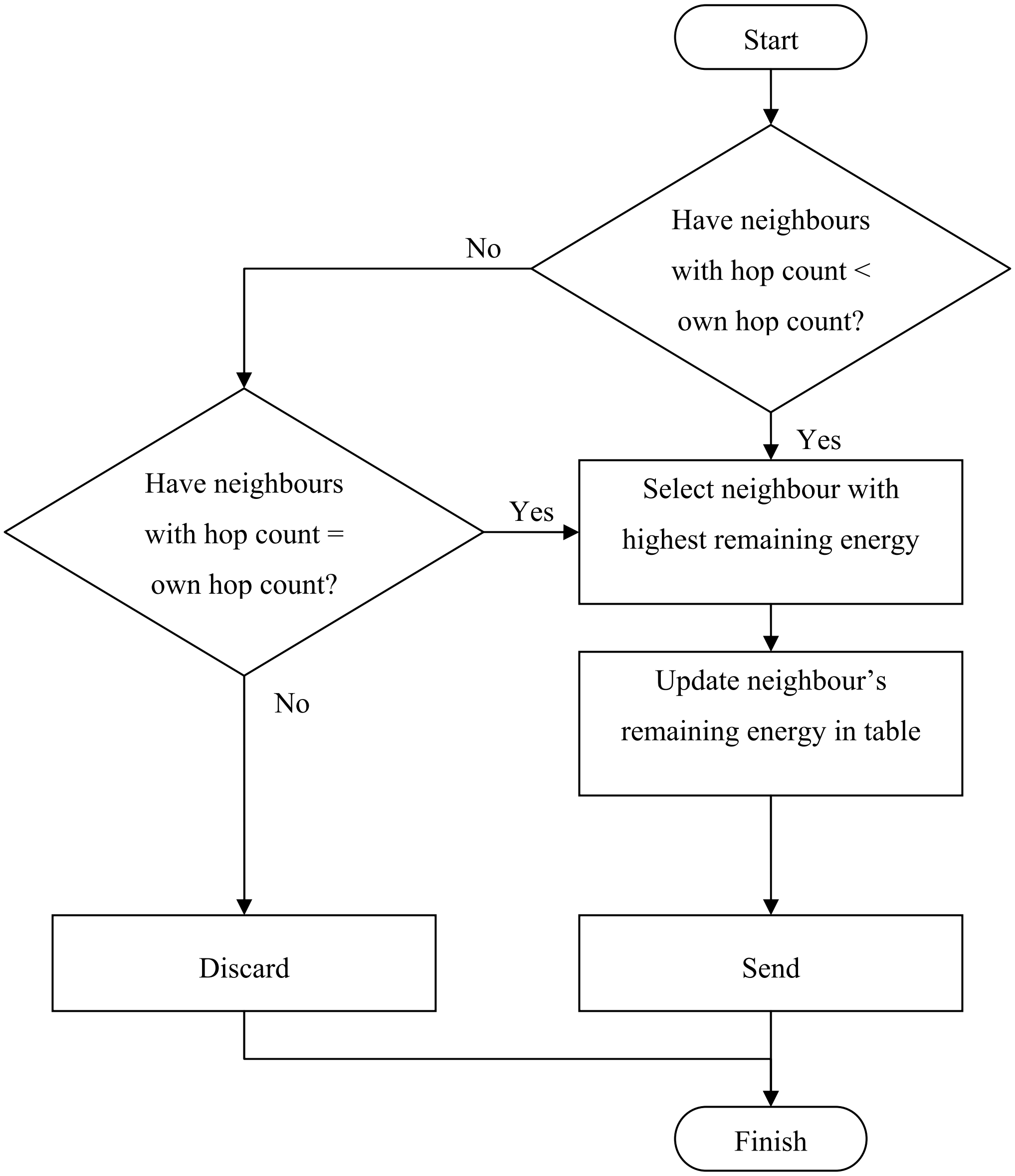
Start
Have neighbours with hop count < own hop count?
No
Yes
Have neighbours with hop count = own hop count?
Yes
No
Select neighbour with highest remaining energy
Update neighbour's remaining energy in table
Discard
Send
Finish

Figure 4.1: Neighbour selection process for sending a new data message.

STEP 3: Forwarding data

When nodes receive a data message they update the remaining energy value in the neighbour table for the neighbour that sent the message. Nodes that forward data messages follow the same process, except for minor differences, that the originating node uses to select the next neighbour in the routing path. The most important difference is that forwarding nodes take the creator address and source address into consideration when selecting the next hop neighbour. When searching the neighbour table for nodes with hop counts smaller or equal to its own, forwarding nodes also make sure that they do not select either the creator of the message, or the node from whom the message was received as the next destination. This ensures that there are no routing loops in the network.

STEP 4: Energy updates

Nodes may be used by more than one neighbour for routing and therefore the energy value stored in the neighbour tables of both of the node's neighbours will not be completely accurate. When a node's remaining energy falls below a certain threshold, it transmits an energy message to all of its neighbours to inform them of its energy level. The fields contained in the header of an energy message are shown in Table 4.3. Energy messages do not contain any data.

Table 4.3: Fields contained in the network layer header of energy messages.

Field	**Size (bits)**
Source address	16
Destination address	16
Hop count	8
Energy level	16
Total	**56**

STEP 5: Network maintenance

The sink node periodically sends a broadcast message through the network so that nodes can add new neighbours that joined the network to neighbour tables and remove neighbours that have failed from the neighbour tables. Nodes also update remaining energy values stored in the neighbour tables. It is important to note that broadcast messages do not contain any data.

The operation of the protocol can be summarised as follows:

1. The sink initialises the network by flooding the network with a broadcast message.
2. Nodes add all their neighbours to their neighbour tables.
3. Nodes send new data along a single path for normal data and along two initial paths for critical data.
4. The neighbour with a hop count that is smaller than the sending node's hop count is selected as the destination.

5. If multiple neighbours have smaller hop counts, the neighbour with the highest remaining energy is selected as the destination.
6. If a node does not have a neighbour with a smaller hop count, it selects the neighbour with the highest remaining energy from neighbours with an equal hop count to it.
7. If the node does not have a neighbour with an equal hop count to it, the message is discarded.
8. Nodes that forward messages select the next hop similarly to originating nodes, but also ensure that the message is not sent to the creator of the message or to the node from whom the message was received.
9. When a node's energy falls below a certain threshold, it sends an energy message to notify its neighbours of its remaining energy.
10. The sink node periodically sends a broadcast message to update and maintain the neighbour tables of the nodes in the network.

CHAPTER 5

Results and Discussion

5.1 Simulation Setup

Simulations of the developed routing protocol were done using the OMNeTT++ Discreet Event Simulation System [37]. The simulator provides a framework for simulating discrete events in networks. Networks and protocols can be modelled using C++ and discrete events can be evaluated using built-in graphical functionality.

To evaluate SEER, it was simulated using OMNeTT++ version 3.1. The network was set up with the sink node in the centre of the network. Nodes were distributed uniformly with each node having up to eight neighbours. Figure 5.1 shows the network layout that was used for simulations.

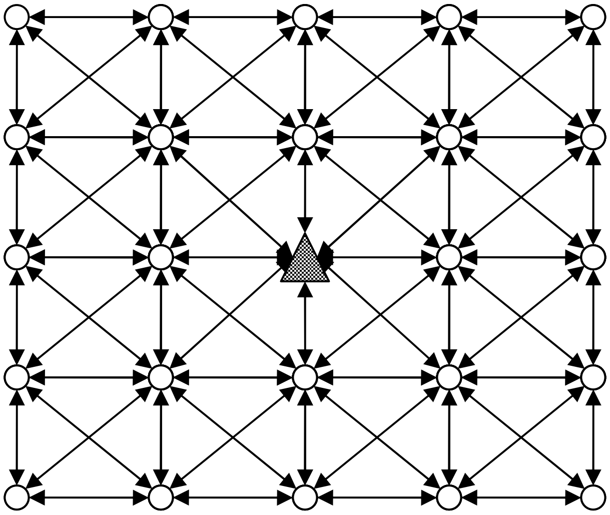

Figure 5.1: An example network of 25 nodes showing layout and connectivity.

The radio model proposed by Heinzelman *et al.* [33] was used to calculate the energy consumed during transmission and reception of messages. According to this model, the energy consumed during transmission (E_{Tx}) is given by:

$$E_{Tx} = E_{elec} \cdot k + \varepsilon_{amp} \cdot k \cdot d^2 \tag{5.1}$$

and the energy consumed during reception (E_{Rx}) is given by:

$$E_{Rx} = E_{elec} \cdot k \tag{5.2}$$

where E_{elec} is the energy consumed by the transceiver electronics, k is the size of the message in bits, ε_{amp} is the energy consumed by the transmitter amplifier and d is the transmission distance in metres. The energy sources of nodes were initialised to 5mJ. 5mJ was used to reduce the simulation time and the required processing by the desktop computer used for the simulations. As in [33], E_{elec} was taken to be 50nJ/bit and ε_{amp} 100pJ/bit. The distance between nodes was assumed to be 1m and nodes were uniformly distributed in the network. No power control mechanism was implemented for the simulations.

SEER was evaluated against three other WSN routing protocols in order to determine its efficiency. *Network lifetime* is defined as the time until the first node fails. The protocols chosen were: *Flooding, directed diffusion* and *SPIN*. Flooding was chosen since it gives an indication of the worst case routing. Directed diffusion was chosen due to the fact that it is very popular in the literature and many protocols have been based on it. SPIN was chosen since it is also a source initiated protocol. The packet sizes for the different packets that were used are given in Table 5.1.

Table 5.1: Packet sizes used for simulations.

Type of message	Size of message (bits)
Data	105
Broadcast	64
Interest	64
Energy	56
Advertise	48
Request	48

The data message consisted of a 73 bit header and 32 bits of data and was used by al four protocols. The header used was the header as discussed in section 4.2. The broadcast message was also used by all four protocols. The interest message was only used by directed diffusion (see section 3.3.3), while the energy message was used only by SEER. The advertise and request messages were used only by SPIN (see section 3.3.2).

The simulation of the protocols started with a broadcast message at the start of the simulation. Every node in the network then sent a new data message every 15 minutes.

For directed diffusion, the first data message, at 15 minutes, was sent along a single path since it was assumed in the simulation that the initial broadcast, at time 0, set up gradients. For subsequent data messages, nodes would flood the messages and upon receiving data messages from nodes, the sink would send a broadcast to set up gradients. Therefore nodes would alternate between flooding and sending along a single path every 15 minutes.

For SPIN, nodes would send an advertisement every 15 minutes and every neighbour would reply with a request message. The originator would then send a data message to every neighbour. These advertisement messages contained an 8 bit sequence number in order to ensure that nodes do not request the same data more than once.

5.2 Simulation Results

Simulations were done using networks consisting of 50, 100, 500, 1000, 1500 and 2000 nodes. This was done to evaluate the scalability of SEER as opposed to the other protocols. Only the results for simulations of networks with 2000 nodes are shown and discussed in this chapter. The results for all of the other network sizes are given in Addendum A at the end of this dissertation.

Simulations were performed to evaluate the network lifetime achieved by each protocol, as well as the number of messages generated by each protocol. The following simulation tests were performed to verify the success of SEER:

Test 1: The time until the first node fails (Figure 5.2).
Test 2: The time until the sink is unreachable, due to all of its neighbours failing (Figure 5.3).
Test 3: The time instant when the number of active nodes in the network reaches a selected percentage (Figure 5.4).
Test 4: The average remaining energy of all the nodes in the network, at selected intervals (Figure 5.5).
Test 5: The average number of data messages sent in the network at selected intervals (Figure 5.6).
Test 6: The number of data messages received by the sink at selected intervals (Figure 5.7).

The results for Tests 1 to 4 were plotted using a ***logarithmic scale*** due to the large improvement achieved by SEER.

5.2.1 Simulations to Evaluate Network Lifetime Performance

The results from Test 1 (Figure 5.2) show that SEER achieves an improvement of several orders of magnitude better than the other protocols tested. This is due to the fact that messages are sent along a single routing path which eliminates energy consuming transmissions. The improvement in network lifetime for the network of 2000 nodes is only three times that of the other protocols, due to the fact that every node in the network sends its data through the nodes surrounding the sink. Therefore, 1999 messages are transmitted by eight nodes every fifteen minutes.

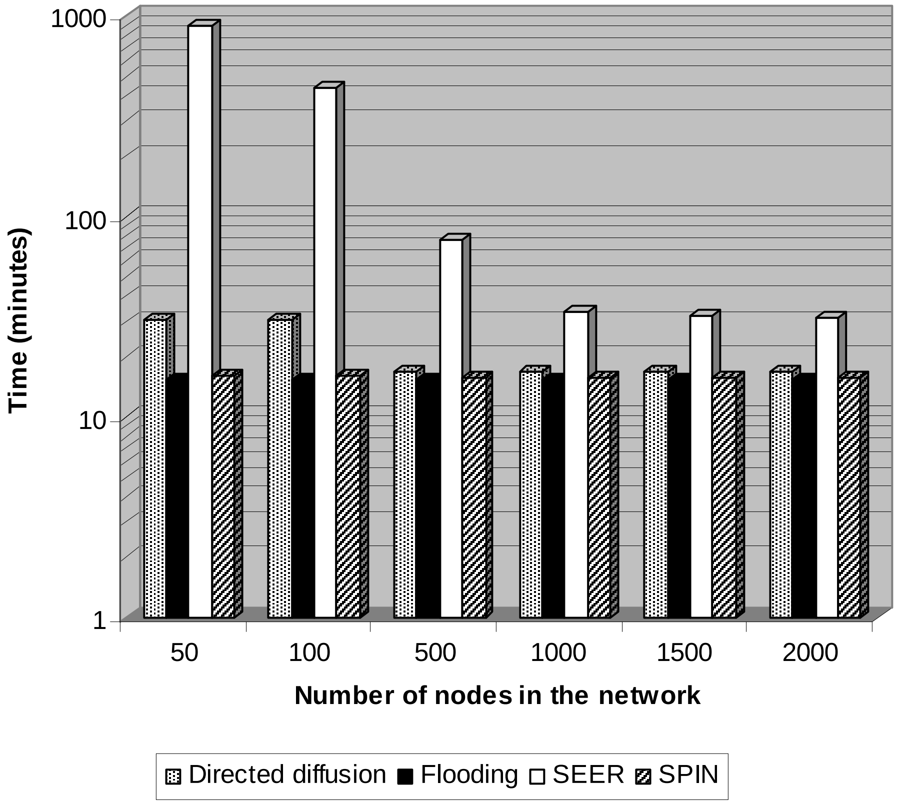

Figure 5.2: Time at which the first node fails due to depleting its energy source.

The results of Test 2 (Figure 5.3) add onto the results of Test 1, indicating that SEER can perform longer before the sink node is unreachable. As the network size increases, the number of messages that have to be routed by the eight nodes surrounding the sink increases and reduces their lifetime. The other protocols cause the network to fail after the first data messages are sent by the nodes at 15 minutes. This is due to the flooding used by all of the protocols.

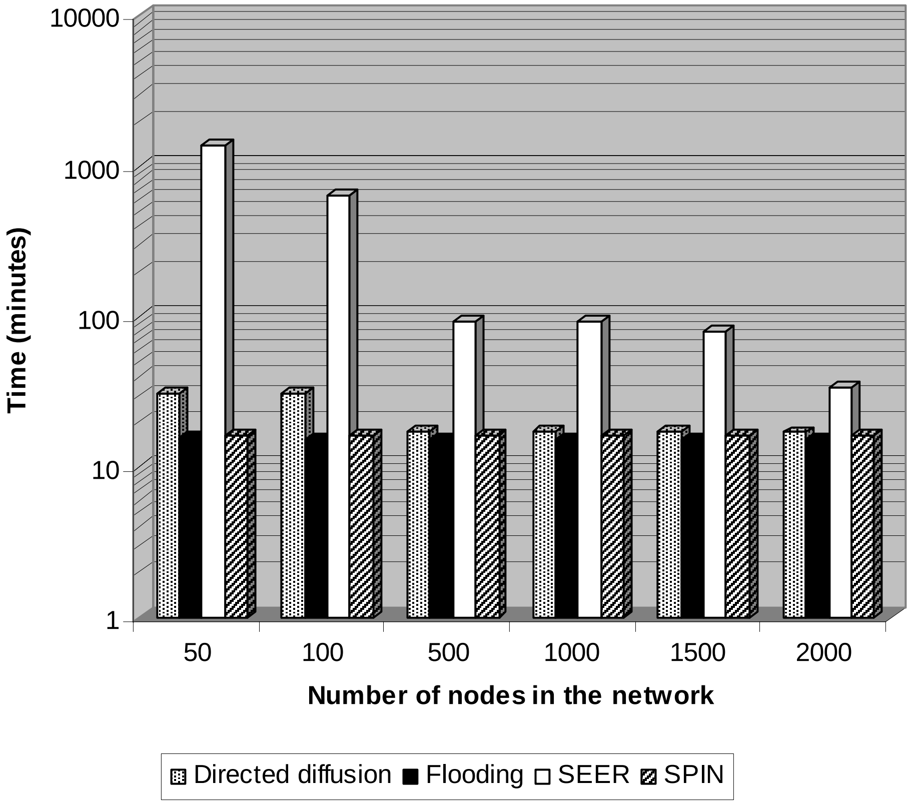

Figure 5.3: The time at which the sink becomes unreachable, due to its last neighbour failing.

Test 3 (Figure 5.4) clearly shows that SEER prolongs the lifetime of the network much more than other flat routing protocols. This large improvement in network lifetime has to do with the fact that the initial energy of all the nodes was set to 5mJ and nodes use 5.1nJ for every message transmitted and 5nJ for every message received. The fact that nodes in the other protocols transmit to every neighbour dramatically increases the energy consumption and consequently reduces lifetime.

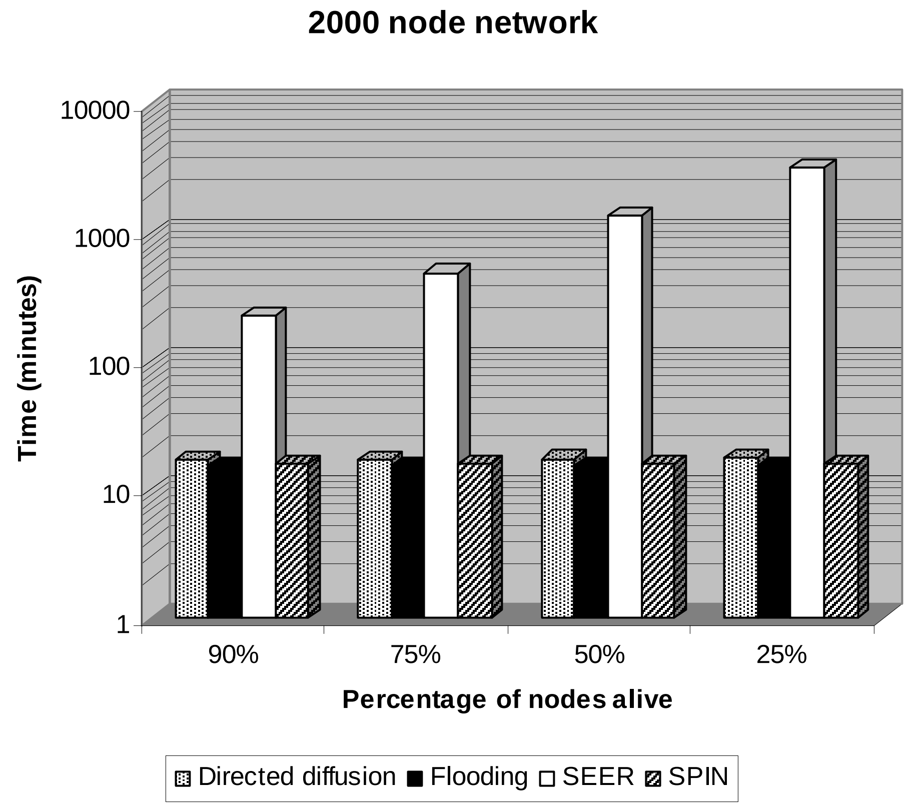

Figure 5.4: The time instant when the number of active nodes in a 2000 node network reaches a selected percentage.

The average remaining energy of all of the nodes in the network is conserved for a much longer time in SEER than in the other protocols, as can be seen from the results of Test 4 (Figure 5.5). This is due to the reduced number of messages that are transmitted. It is important to note that the energy level of SEER does drop even though it is not clear on the graph in Figure 5.5. This is due to the fact that the scale of the y-axis is logarithmic and the increments are 1,000,000nJ.

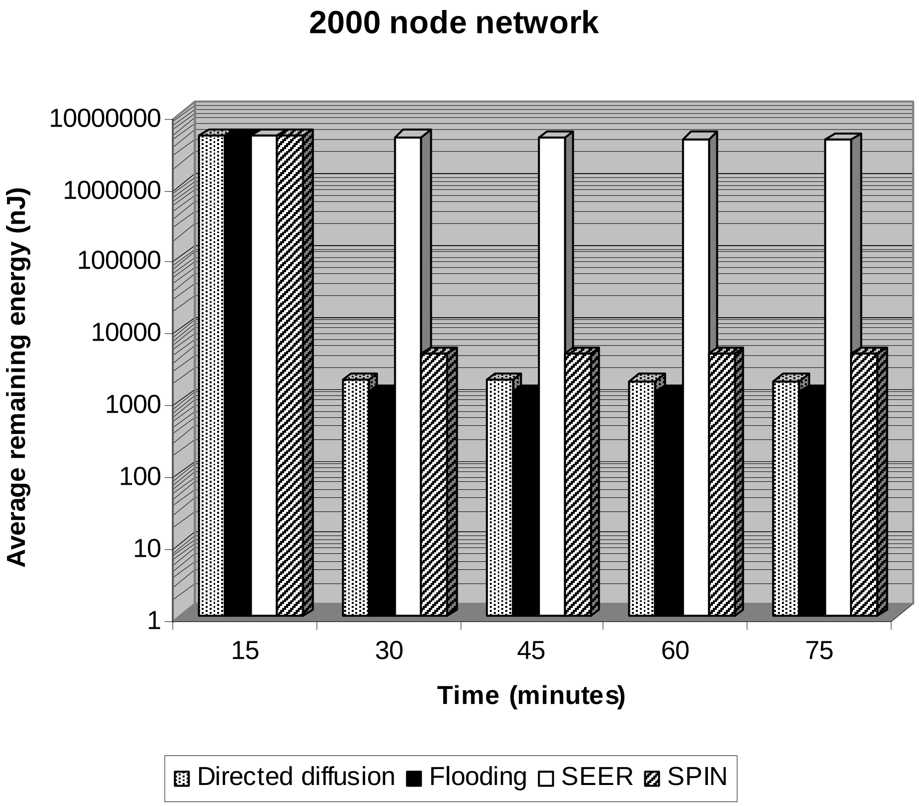

Figure 5.5: The average remaining energy of all the nodes in a 2000 node network at selected intervals.

5.2.2 Simulations to Evaluate the Number of Messages Generated

Results from Test 5 (Figure 5.6) show that for the worst case, SEER nodes on average need to transmit only about a quarter of the number of messages that nodes in the best competitor have to transmit. The number of messages that SPIN transmits is much more than the other protocols due to the fact that each node broadcasts an advertisement and then sends data to all of its neighbours that sent a request.

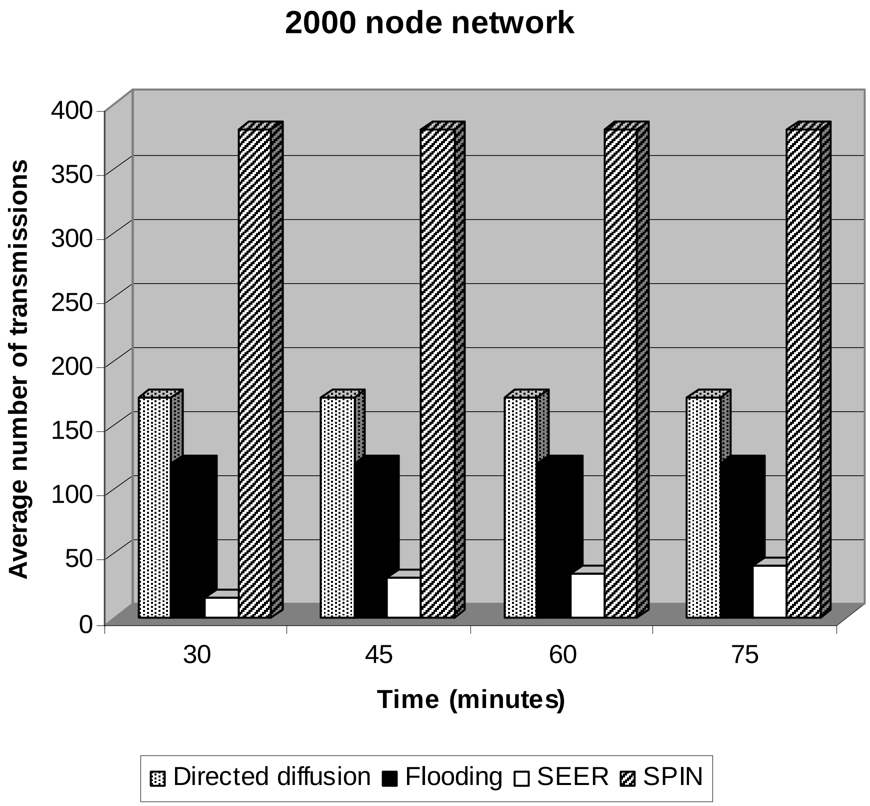

Figure 5.6: The average number of data messages sent in a 2000 node network at selected intervals.

Test 6 (Figure 5.7) indicates that the sink does not receive a data message for every node in the network for any of the other three protocols, due to the fact that the nodes surrounding the sink deplete their energy before all the messages can reach the sink. In contrast to the other three protocols, the figure shows that at 30 minutes the sink has received one data message for every node in the network for the SEER protocol. In the SEER scenario, the nodes surrounding the sink deplete their energy just after 30 minutes and therefore no more messages reach the sink after 45 minutes.

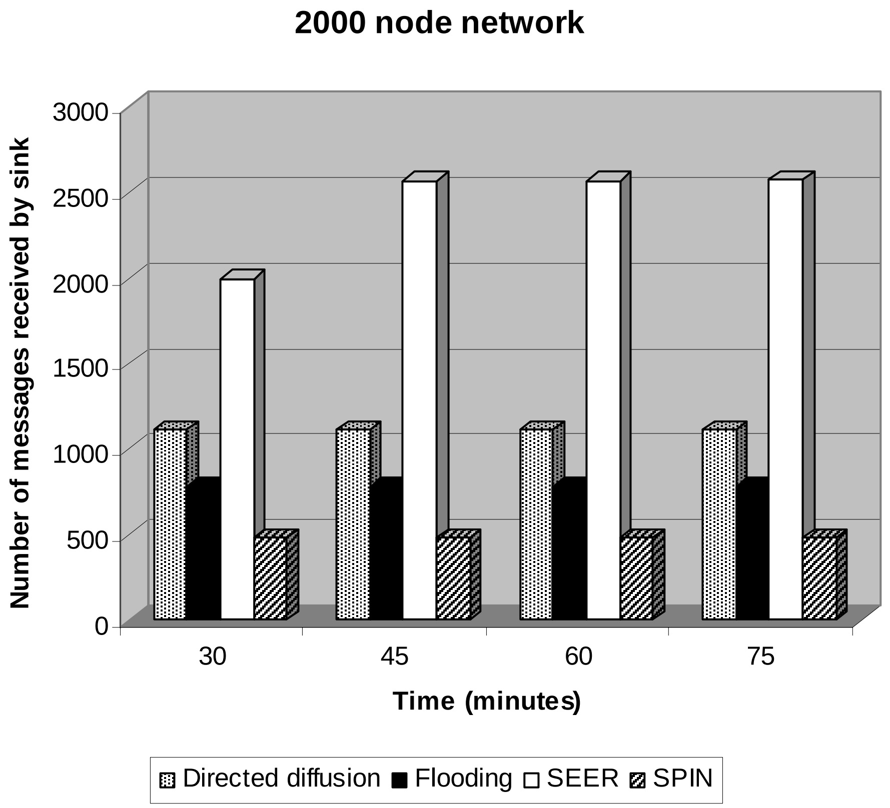

Figure 5.7: The number of data messages received by the sink in a 2000 node network at selected intervals.

The results clearly show that the node failure rate increases the closer nodes are to the sink. This causes a problem for a WSN since the nodes surrounding the sink fail much sooner than nodes far away from the sink. This means that data from the nodes that are still active cannot reach the sink. A possible solution to this problem is to increase the node density as the distance to the sink decreases. The results from the six tests confirm that SEER scales well and improves network lifetime by limiting the number of messages that are sent through the network. Please refer to Addendum A for further simulation results.

CHAPTER 6

Conclusion

Networks of small, inexpensive, disposable, smart sensors are emerging as a new technology with tremendous potential. Wireless sensor networks can be randomly deployed inside or close to phenomenon to be monitored. The advantage of these networks is the fact that they are self-configuring, which means that a sensor network can be deployed randomly on a battlefield, in a disaster area or in an inaccessible area without the need for human intervention.

The energy supplies of nodes are not replenished or replaced and therefore nodes only participate in the network for as long as they have energy. This fact necessitates energy efficiency in the design of every aspect of such nodes. Energy consumption in sensor nodes occurs mainly due to computational processing and, to a greater extent, communication. The routing protocol employed by these sensor nodes can minimize the number of transmissions that nodes make as well as the computational complexity of routing path selection. It is therefore of critical importance that the routing protocol be designed with energy efficiency in mind.

A new wireless sensor network routing protocol was developed and evaluated during this research. *Simple Energy Efficient Routing (SEER)* fills the void that exists in current literature for an energy efficient routing protocol that is not dependent on sensor node hardware or sensor network layout.

The development of the protocol required a thorough study of the specific requirements of WSNs as well as an evaluation of current proposals for WSN routing protocols. The study showed that energy conservation in a WSN node can be accomplished by minimizing the number of transmit operations and the computational processing that is required from the node. WSN routing protocols can be classified as either flat or hierarchical, according to the network structure. It was found that flat protocols are computationally less complex and more scalable than hierarchical protocols. Data transmission in a WSN is either source initiated or destination initiated. The study found that source initiated protocols require less messages than destination initiated protocols, making it more energy efficient.

The SEER protocol was therefore designed to use a *flat network structure* and *source initiated* communication with *event-driven reporting*. Routing decisions are based on two routing metrics: *hop count* and *remaining energy*. A node routes a data message to a neighbouring node that has a hop count less than or equal to its own hop count. If more than one neighbour exists that satisfy this requirement, the neighbour with the highest remaining energy is chosen as the destination for the message. This simple method for choosing the next hop in a multi-hop routing path is novel and leads to enormous reductions in energy consumption.

Simulation results showed that SEER increases network lifetime dramatically compared to other protocols. This increase is a direct result of the decrease of the number of messages that nodes have to transmit. The protocol does not impose any hardware requirements and was shown to be scalable.

The protocol does not have any built-in reliability. This is due to the fact that energy efficiency is paramount to WSNs and handshaking or multiple paths would increase message overhead. Reliability in WSNs is achieved by the number of nodes. Due to the large number of nodes, it is very likely that more than one node will observe an event and therefore transmit a message to the sink. The neighbour tables that each node has to keep are proportional to the number of neighbours that it can communicate with. Nodes that are able to communicate with a large number of neighbours can reduce their transmission power and thereby gain even more energy savings.

Future research efforts might focus on the layout of the network. The results obtained during this research shows that for a network with uniform node distribution, the node failure rate increases as the distance to the sink decreases. This causes the whole network to loose connectivity as soon as the nodes surrounding the sink fail. If the density of nodes around the sink is increased, the connectivity will be conserved longer. Another area for future research is efficient locationing. The sink node needs to know where the data that it receives is coming from. One option is for nodes to be manually deployed and their locations to be manually entered into a database on the sink. This is not effective for large sensor networks. Hardware such as GPS modules increase energy consumption and cannot be used indoors. An efficient method for establishing relative or exact locations for nodes is essential for large self-configuring WSNs.

Overall, the routing protocol developed during this research is novel and makes an important contribution to the literature by being simple enough to be physically implemented on a variety of existing WSN nodes while still achieving a very high level of energy efficiency.

References

[1] “21 ideas for the 21st century,” *Business Week*, pp. 78–167, Aug. 30, 1999.

[2] “10 emerging technologies that will change the world,” *Technology Review*, vol. 106, no. 1, pp. 33–49, Feb. 2003.

[3] K. Intae and R. Poovendran, “Maximizing static network lifetime of wireless broadcast ad hoc networks*”,* in *Proceedings of the IEEE International Conference on Communications, 11-15 May 2003, Anchorage, USA*, vol. 3, 2003, pp. 2256 – 2261.

[4] W. Heinzelman, J. 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 Aug. 1999, Seattle, USA*, 1999, pp. 174–185.

[5] 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 Aug. 2000, Boston, USA*, 2000, pp. 56–67.

[6] D. Braginsky and D. Estrin, “Rumor Routing Algorithm for Sensor Networks,” in *Proceedings of the first Workshop on Sensor Networks and Applications, 28 Sept. 2002, Atlanta, USA*, 2002, pp. 22–31.

[7] F. Ye , A. Chen, S. Lu and L. Zhang “A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks,” in *Proceedings of the IEEE International Conference on Computer Communication and Networks (ICCCN), 15-17 Oct. 2001, Phoenix, USA*, 2001, pp. 304–309.

[8] C. Schurgers and M.B. Srivastava, “Energy efficient routing in wireless sensor networks”, in *Proceedings of the IEEE Military Communications Conference (MILCOM), 28-31 Oct. 2001, Washington, USA*, vol. 1, 2001, pp. 357–361 .

[9] R. C. Shah and J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference, 17-21 Mar. 2002, Orlando, USA*, vol. 1, 2002, pp. 350–355.

[10] V. Rodoplu and T. H. Meng, "Minimum Energy Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, Aug. 1999.

[11] S. Servetto and G. Barrenechea, "Constrained Random Walks on Random Graphs: Routing Algorithms for Large Scale Wireless Sensor Networks," in *Proceedings of the first International Workshop on Wireless Sensor Networks and Applications, 28 Sept. 2002, Atlanta, USA*, 2002, pp. 12–21.

[12] L. Li, and J. Y. Halpern, "Minimum-Energy Mobile Wireless Networks Revisited," in *Proceedings of the IEEE International Conference on Communications (ICC), 11-15 Jun. 2001, Helsinki, Finland*, 2001, vol. 1, pp. 278–283.

[13] C. Chong and S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," in *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[14] S. Mahlknecht, "WSSN (Wireless Self-sustaining Sensor Network) Project," 2005, http://www.ict.tuwien.ac.at/wireless/. Last accessed on 18 April 2005.

[15] S. Mahlknecht and M. Bock, "CSMA-MPS: a minimum preamble sampling MAC protocol for low power wireless sensor networks," in *Proceedings of the IEEE International Workshop on Factory Communication Systems (IWFC), 22-24 Sept. 2004, Vienna, Austria*, 2004, pp. 73–80.

[16] University of California, Berkeley, "A spec of smart dust," 2003, http://www.coe.berkeley.edu/forefront/fall2003/breakthroughs.html. Last accessed on 3 August 2005.

[17] W. Stallings, *Data & Computer Communications*, 6th ed., Prentice Hall, New Jersey, p. 453, 2000.

[18] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks (Elsevier),* vol. 38, pp. 393-422, 2002.

[19] G. Karayannis, "Emerging Wireless Standards: Understanding the Role of IEEE 802.15.4 & ZigBee in AMR & Submetering," 2003, http://www.zigbee.org/imwp/idms/popups/pop_download.asp?contentID=820. Last accessed on 21 April 2005.

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

[21] 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)*, *Jul. 2001, Ambleside, UK*, 2001, pp. 1–6.

[22] G. J. Pottie and W. J Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 551–558, May 2000.

[23] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Mobile networking for smart dust," in *Proceedings of the 5th International Conference on Mobile Computing and Networking (Mobicom), 15-19 Aug. 1999, Seattle, USA*, 1999, pp. 271–278.

[24] 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), 16-21 Jul. 2001, Rome, Italy*, 2001, pp. 272–287.

[25] Bluetooth SIG, "Bluetooth Specification v1.1," 2001, http://www.bluetooth.org/spec/. Last accessed on 20 April 2005.

[26] L. D. Paulson, "Will ultrawideband technology connect in the marketplace?," Computer, vol. 36, issue 12, Dec. 2003, pp. 15–17.

[27] IEEE 802.15 WPAN Task Group 4, "IEEE 802.15.4 Standard 2003," 2003, http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf. Last accessed on 20 April 2005.

[28] M. Ilyas, *The Handbook of Ad Hoc Wireless Networks*, CRC Press, Boca Raton, USA, pp. 14-2 to 14-3, 2003.

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

[30] A. El-Hoiydi, "Spatial TDMA and CSMA with Preamble Sampling for Low Power Ad Hoc Wireless Sensor Networks", *in Proceedings of the IEEE Symposium on Computers and Communications (ISCC), 1-4 July 2002, Taormina/Giardini Naxos, Italy*, 2002, pp. 685–692.

[31] Q. Jiang and D. Manivannan, "Routing protocols for sensor networks", in *Proceedings of the 1st IEEE Consumer Communications and Networking Conference, 5-8 Jan. 2004, Las Vegas, USA*, 2004, pp. 93–98.

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

[33] 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), 4-7 Jan. 2000, Hawaii, USA*, 2000, pp. 1–10.

[34] N. Thepvilojanapong, Y. Tobe and K. Sezaki, "HAR: Hierarchy-Based Anycast Routing Protocol for Wireless Sensor Networks," in *Proceedings of the 2005 Symposium on Applications and the Internet, 31 Jan. - 4 Feb. 2005, Trento, Italy*, 2005, pp. 204–212.

[35] M. Hempel, H. Sharif and P. Raviraj, "HEAR-SN: A New Hierarchical Energy-Aware Routing Protocol for Sensor Networks," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS), 3-6 Jan. 2005, Hawaii, USA*, 2005, pp. 324a–324a.

[36] H.-S. Kim and K.-J. Han, "A Power Efficient Routing Protocol Based on Balanced Tree in Wireless Sensor Networks," in *Proceedings of the 1st International Conference on Distributed Frameworks for Multimedia Applications, 6-9 Feb. 2005, Besançon, France*, 2005, pp. 138–143.

[37] OMNeTT++ Discreet Event Simulation System, 2005, http://www.omnetpp.org. Last accessed on 4 April 2005.

ADDENDUM A

Further Results

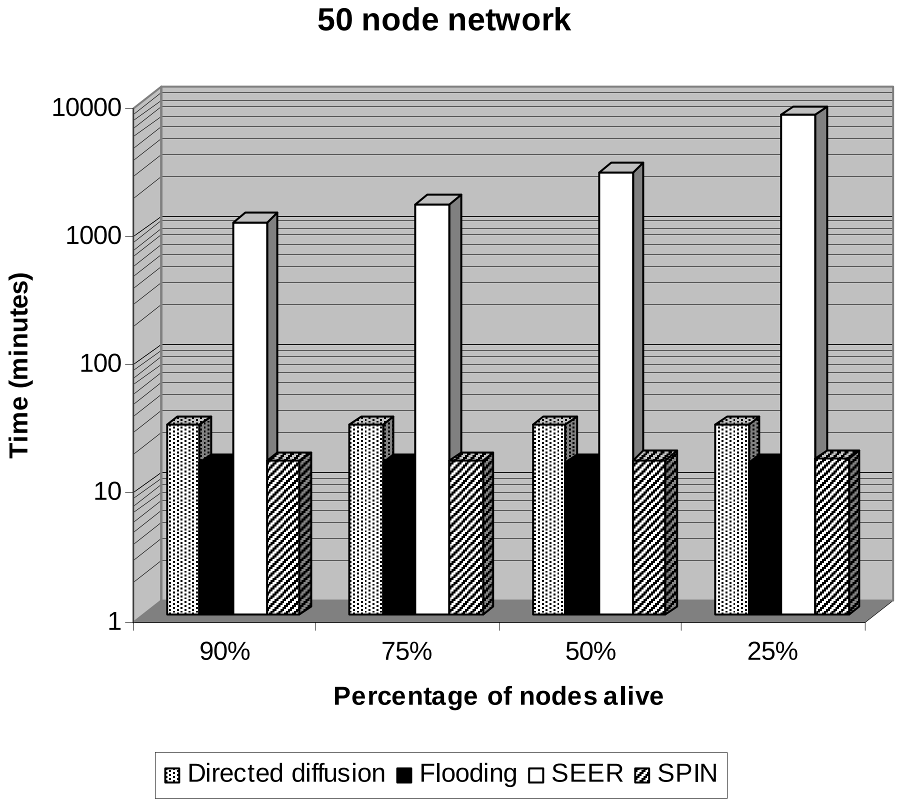

Figure A.1: The time instant when the number of active nodes in a 50 node network reaches a selected percentage.

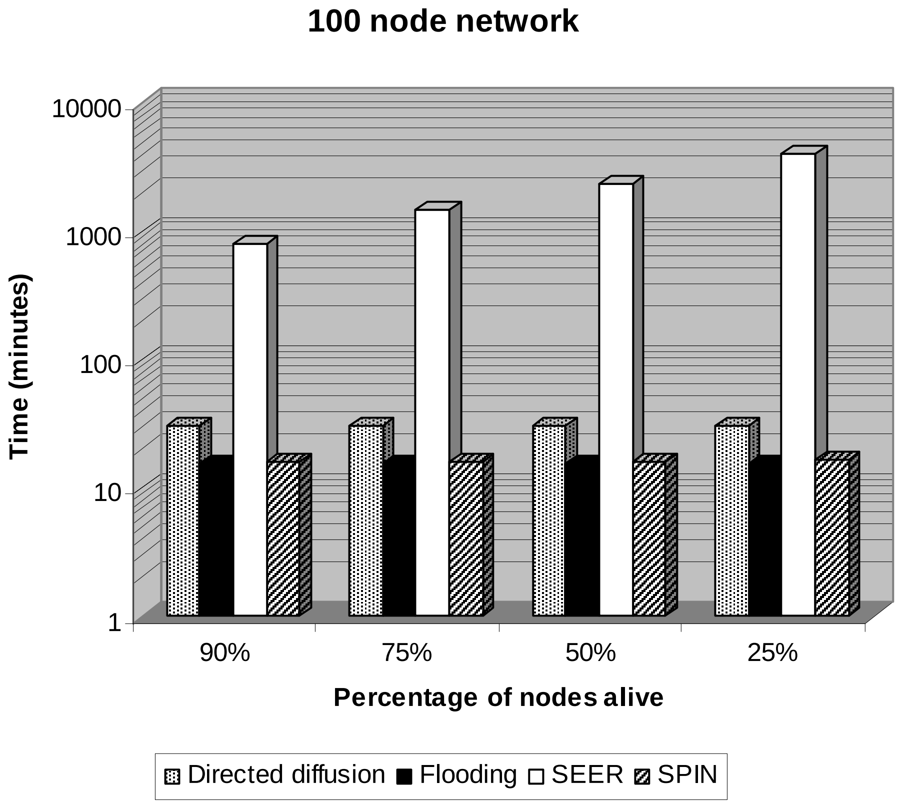

Figure A.2: The time instant when the number of active nodes in a 100 node network reaches a selected percentage.

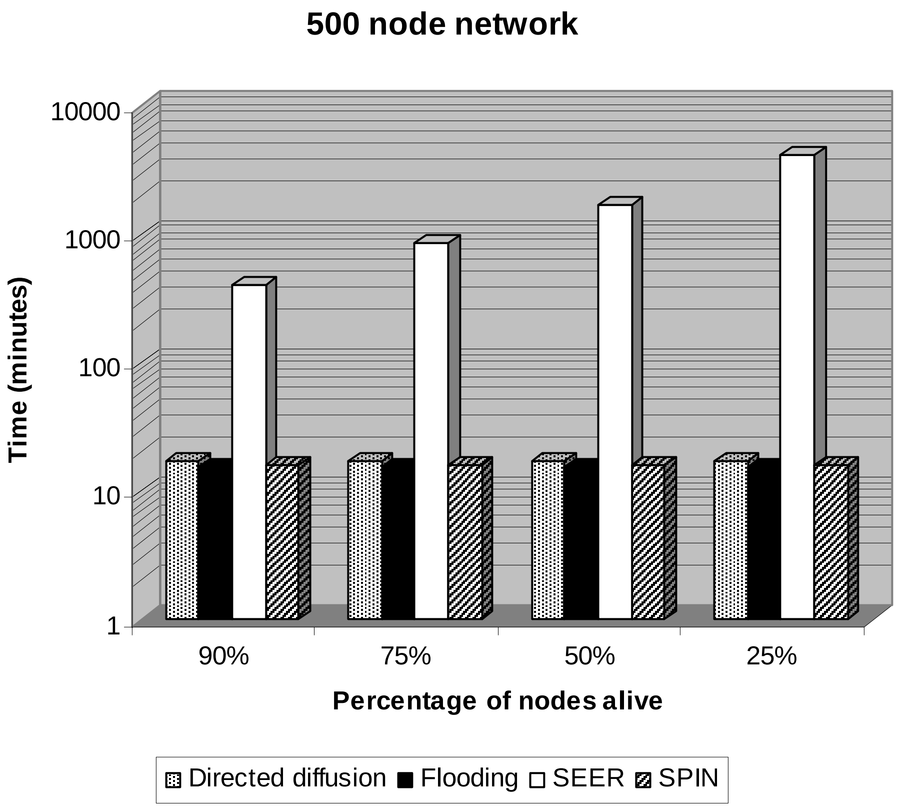

Figure A.3: The time instant when the number of active nodes in a 500 node network reaches a selected percentage.

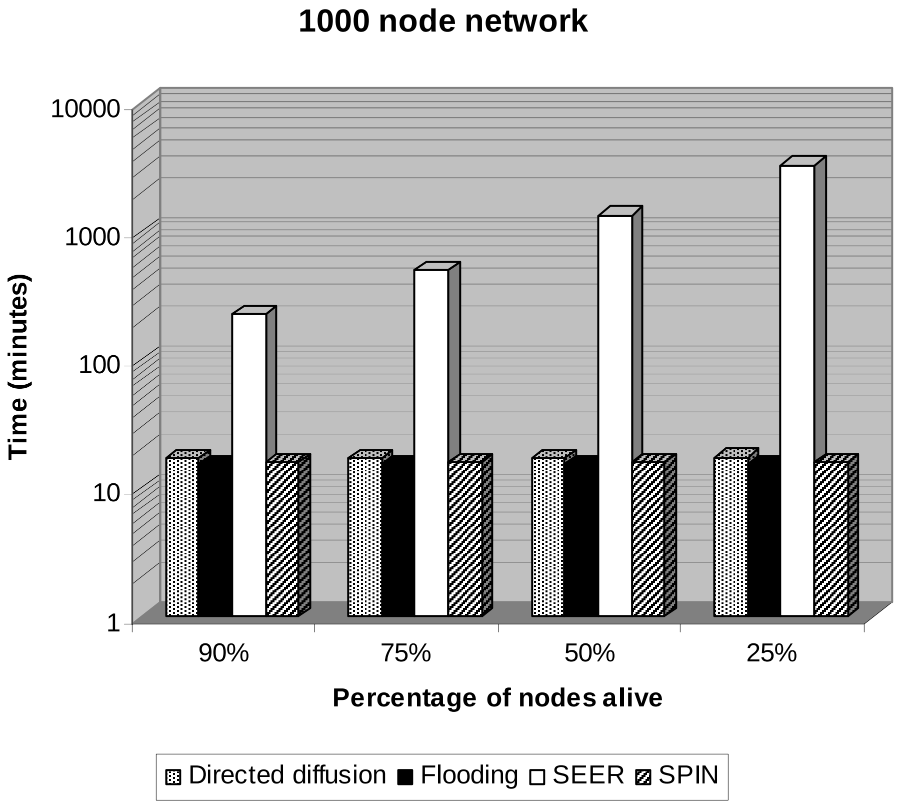

Figure A.4: The time instant when the number of active nodes in a 1000 node network reaches a selected percentage.

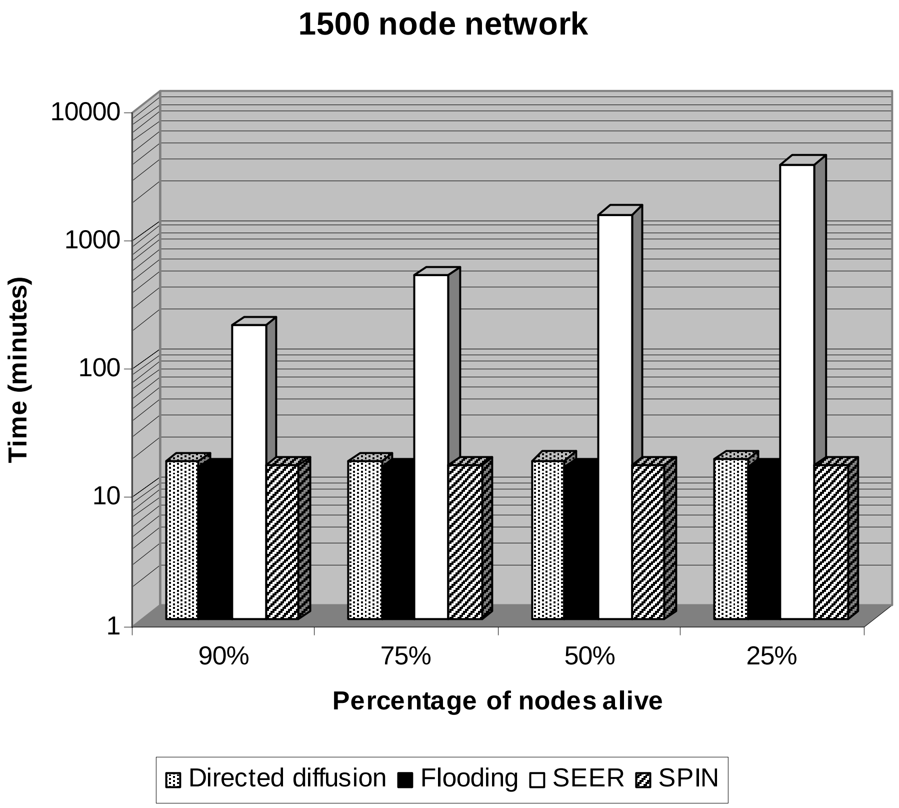

Figure A.5: The time instant when the number of active nodes in a 1500 node network reaches a selected percentage.

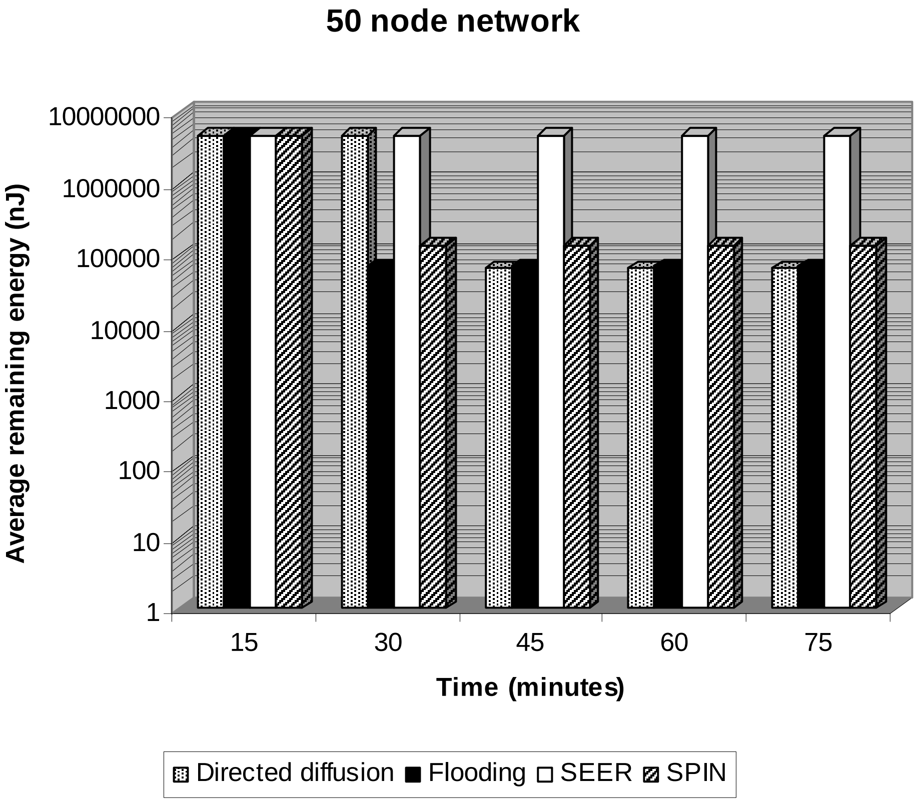

Figure A.6: The average remaining energy of all the nodes in a 50 node network at selected intervals.

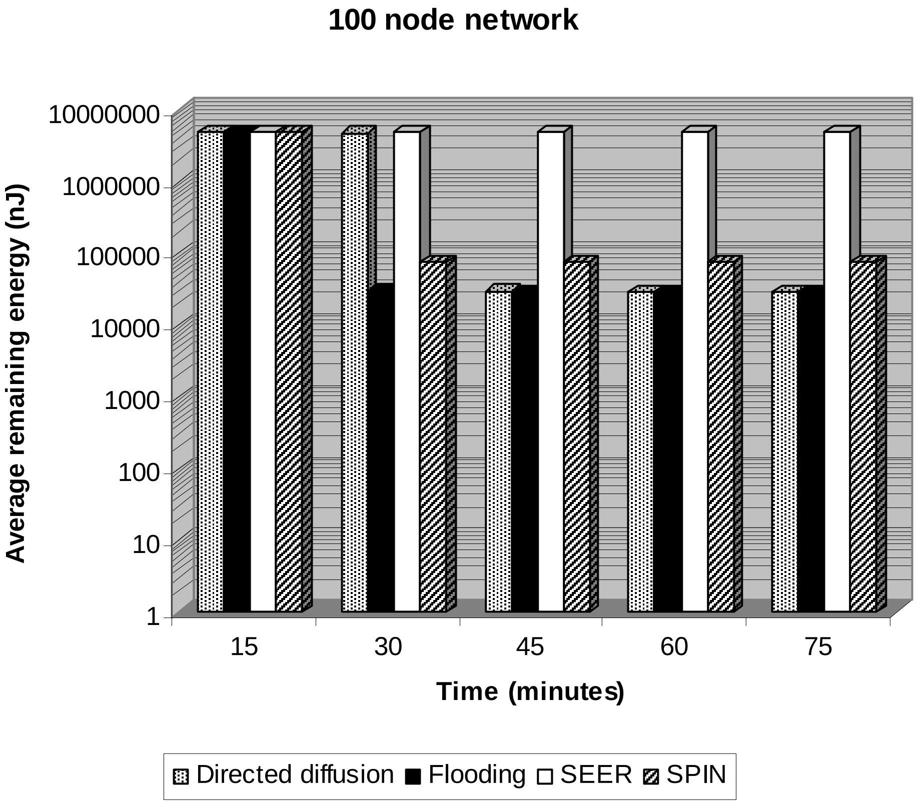

Figure A.7: The average remaining energy of all the nodes in a 100 node network at selected intervals.

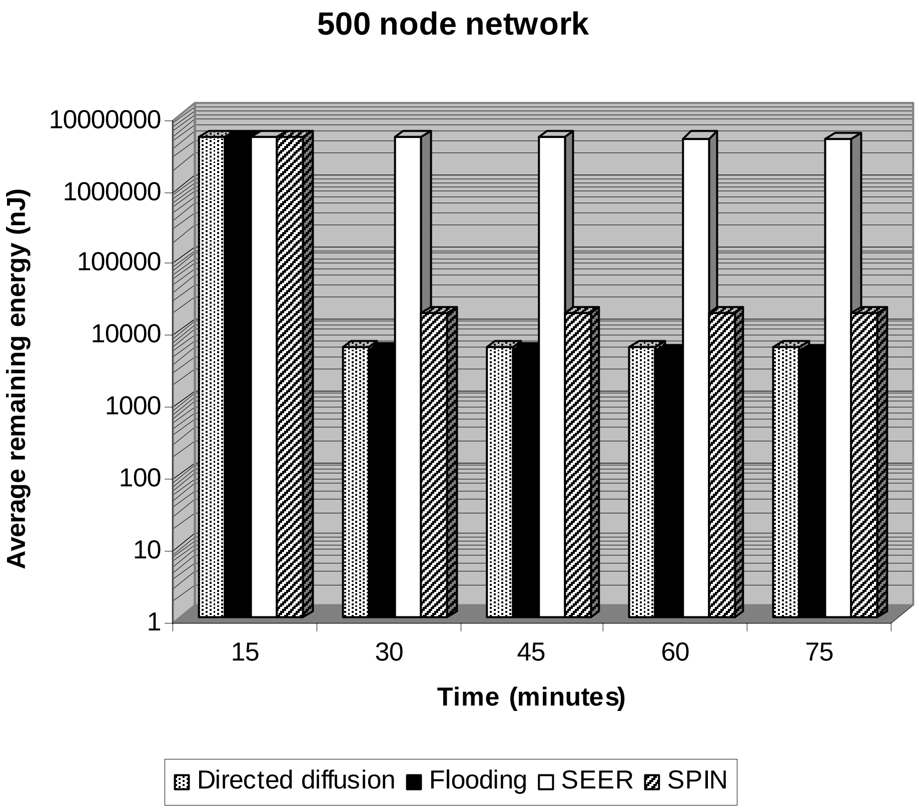

Figure A.8: The average remaining energy of all the nodes in a 500 node network at selected intervals.

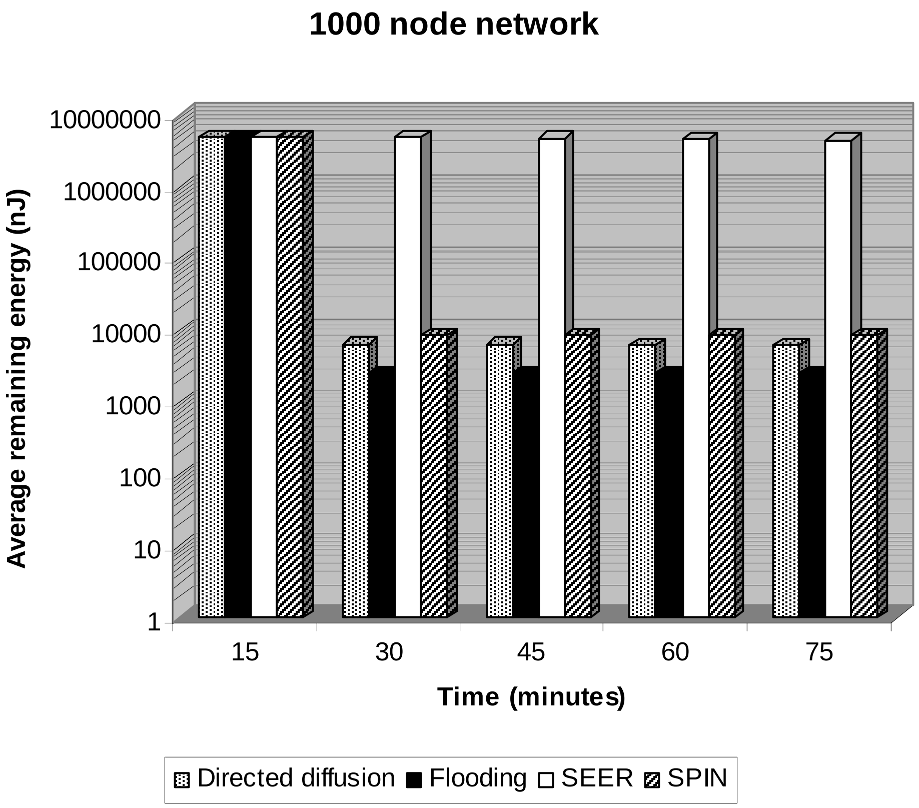

Figure A.9: The average remaining energy of all the nodes in a 1000 node network at selected intervals.

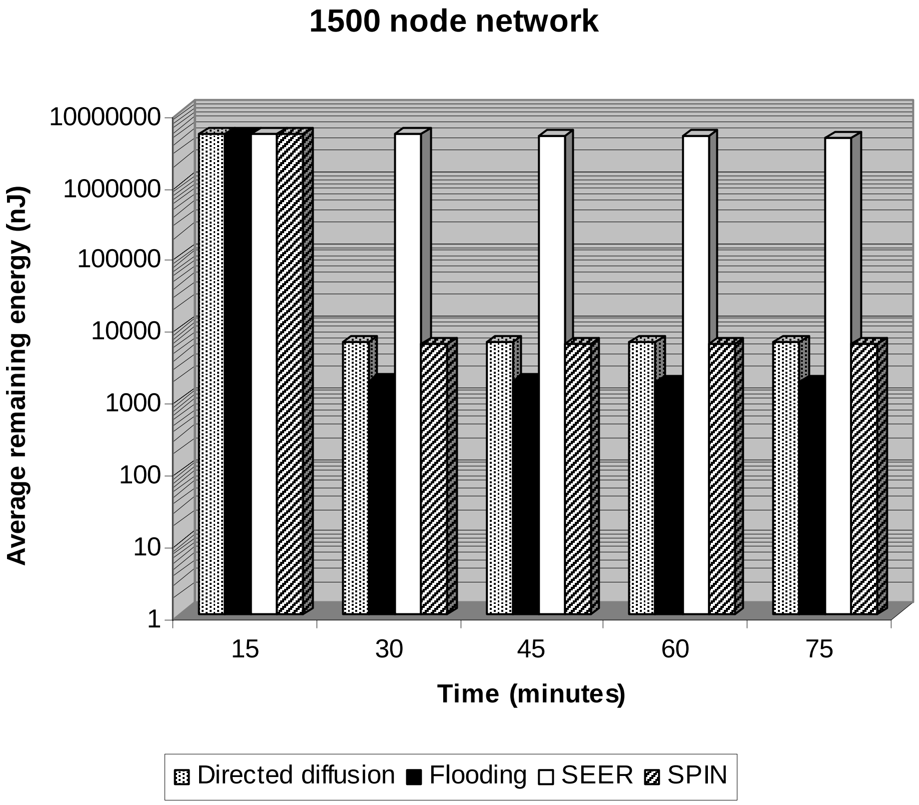

Figure A.10: The average remaining energy of all the nodes in a 1500 node network at selected intervals.

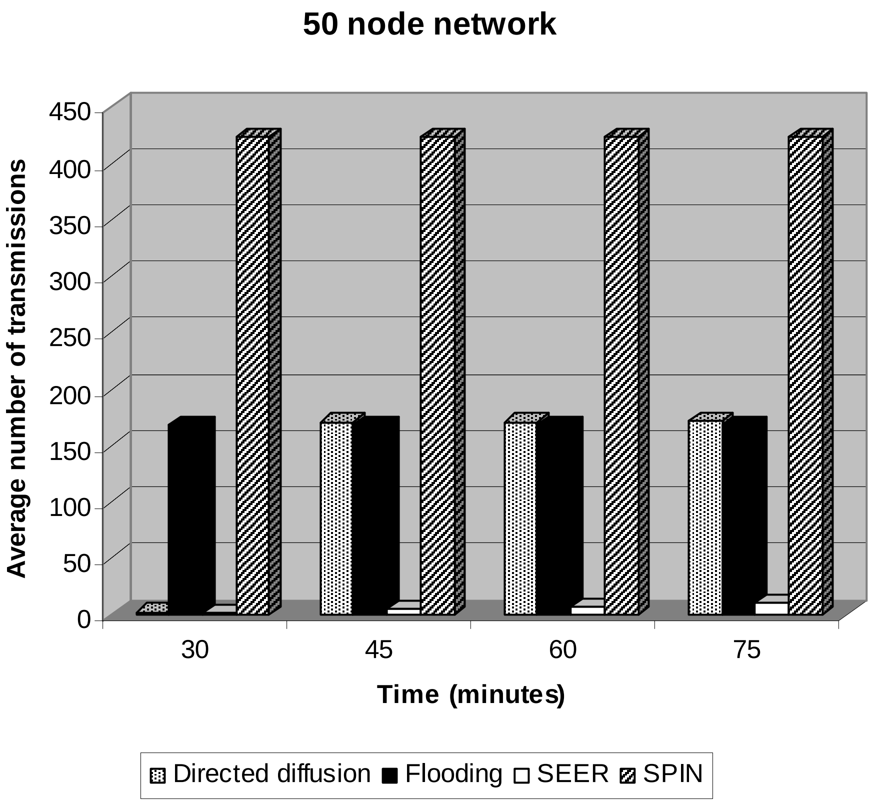

Figure A.11: The average number of data messages sent in a 50 node network at selected intervals.

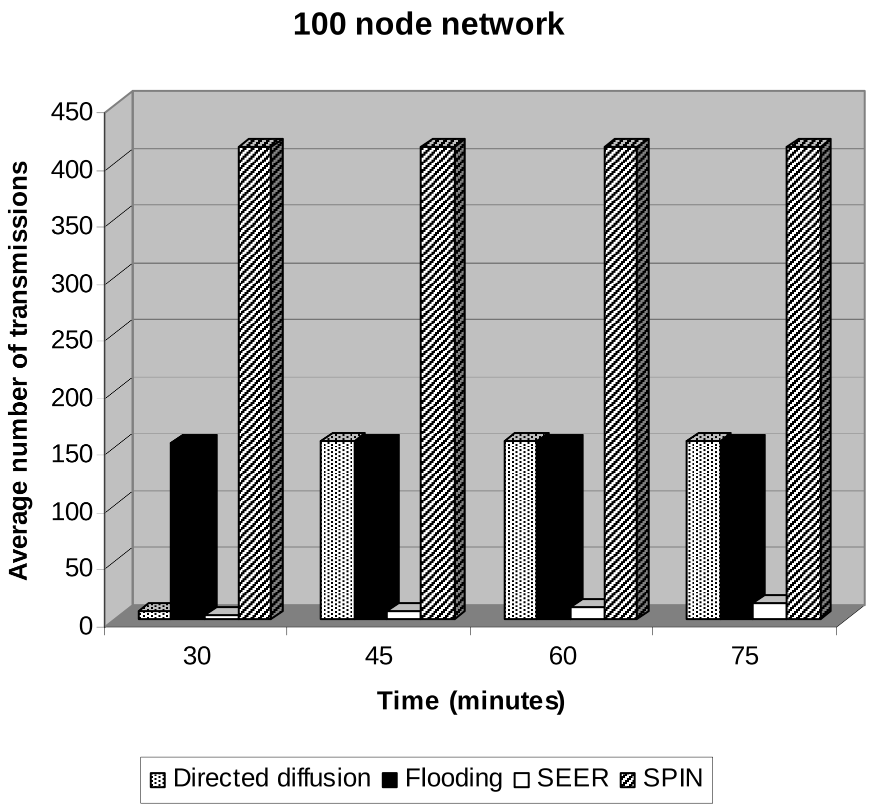

Figure A.12: The average number of data messages sent in a 100 node network at selected intervals.

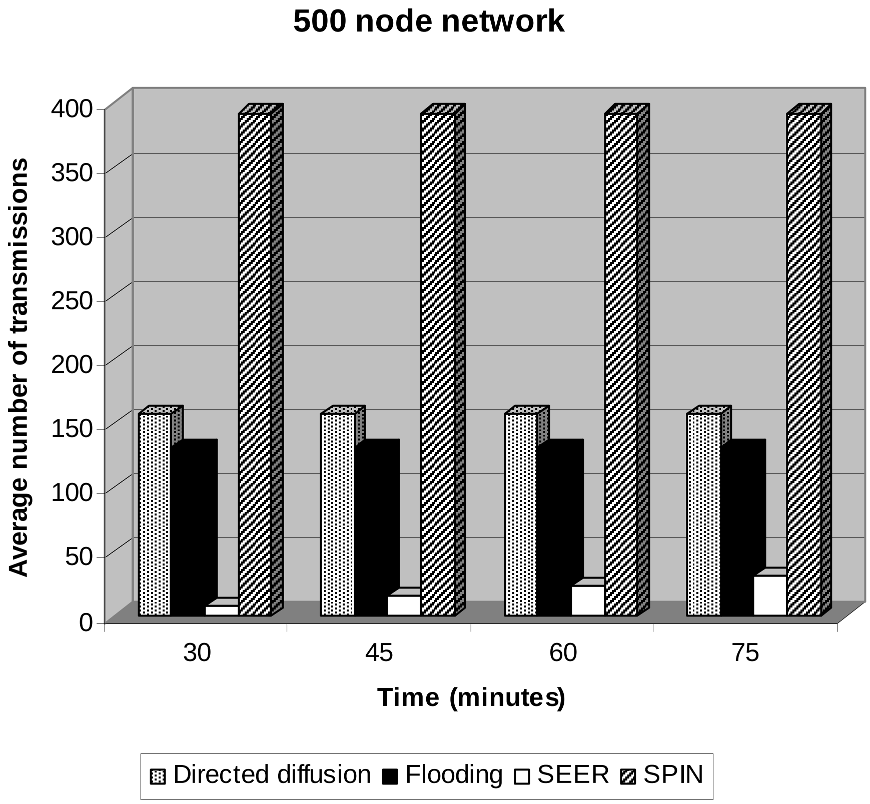

Figure A.13: The average number of data messages sent in a 500 node network at selected intervals.

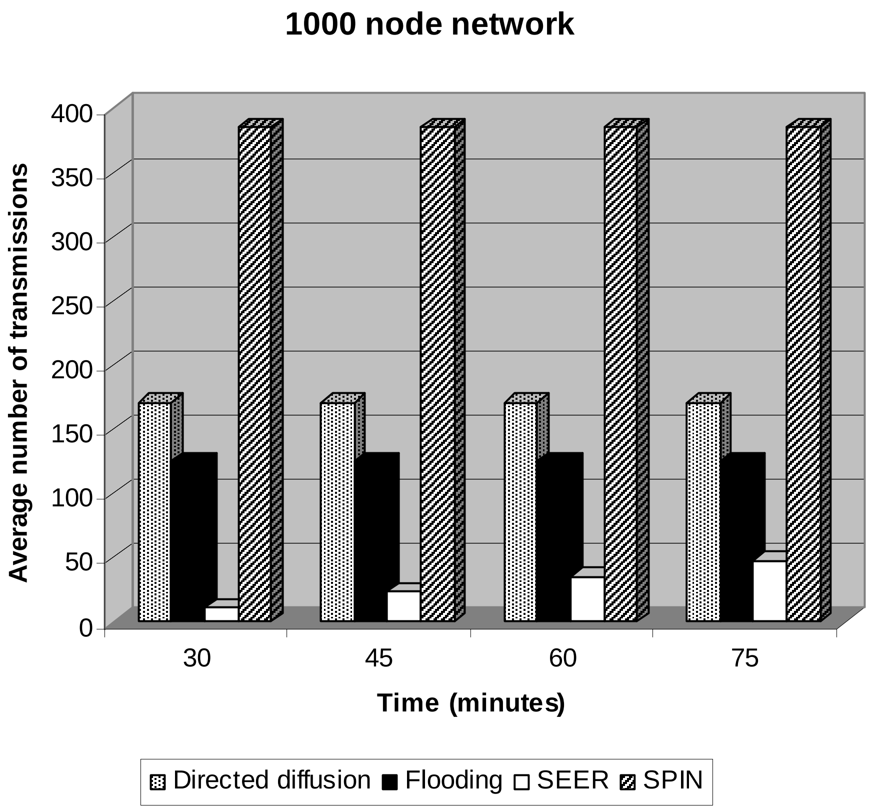

Figure A.14: The average number of data messages sent in a 1000 node network at selected intervals.

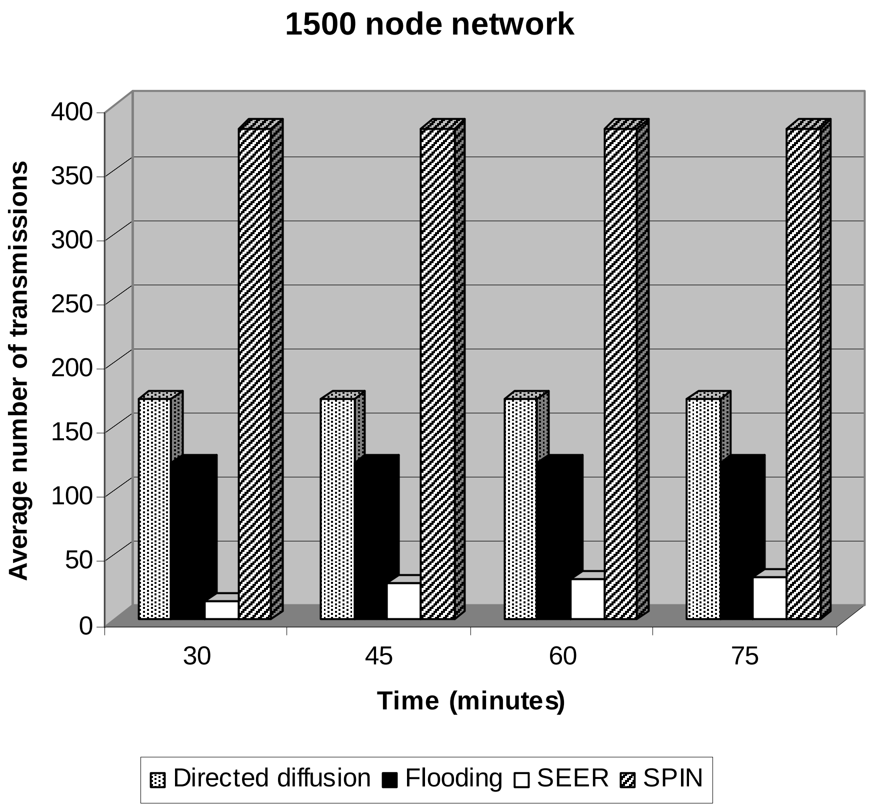

Figure A.15: The average number of data messages sent in a 1500 node network at selected intervals.

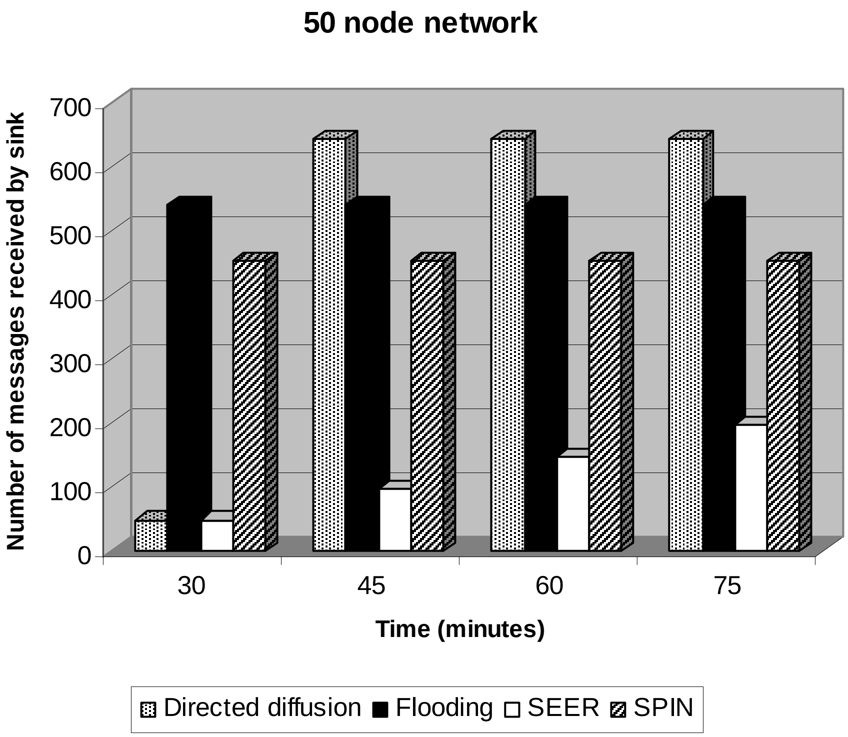

Figure A.16: The number of data messages received by the sink in a 50 node network at selected intervals.

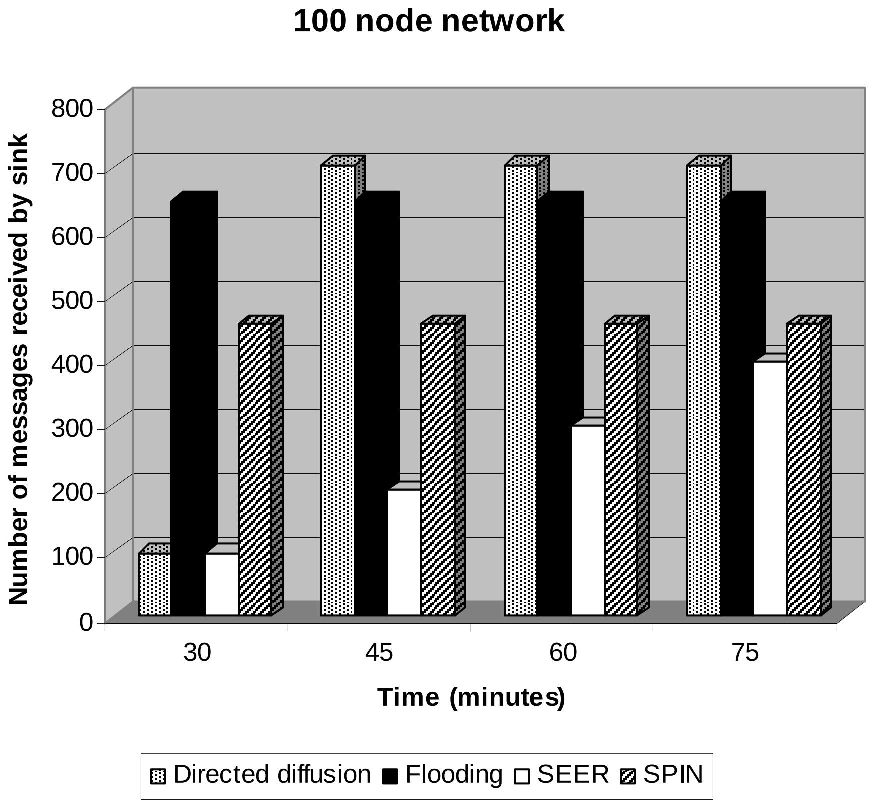

Figure A.17: The number of data messages received by the sink in a 100 node network at selected intervals.

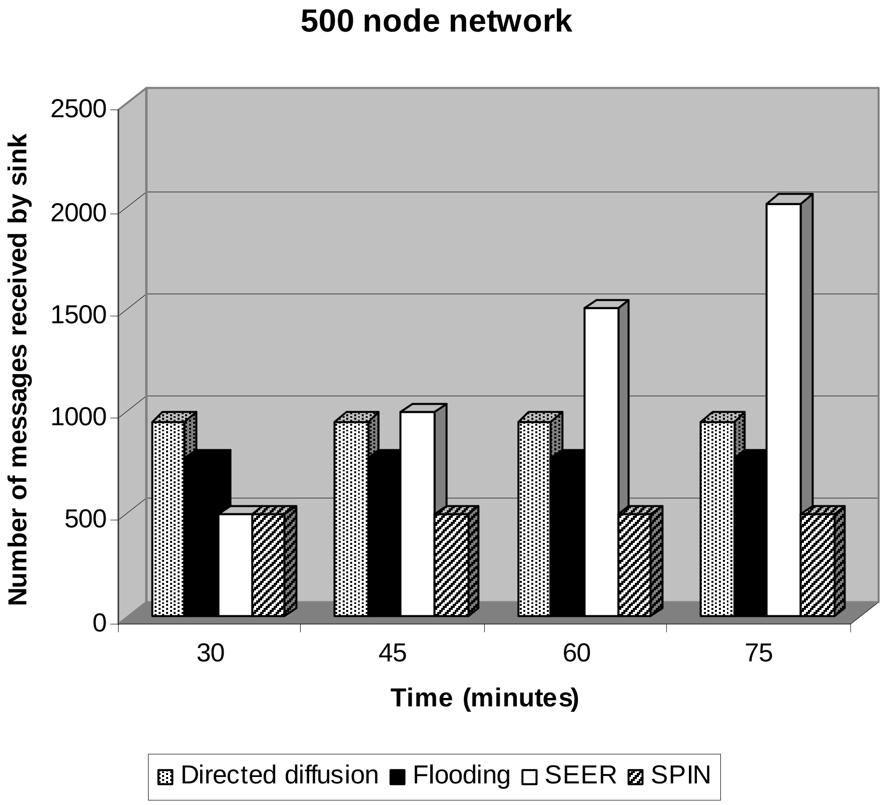

Figure A.18: The number of data messages received by the sink in a 500 node network at selected intervals.

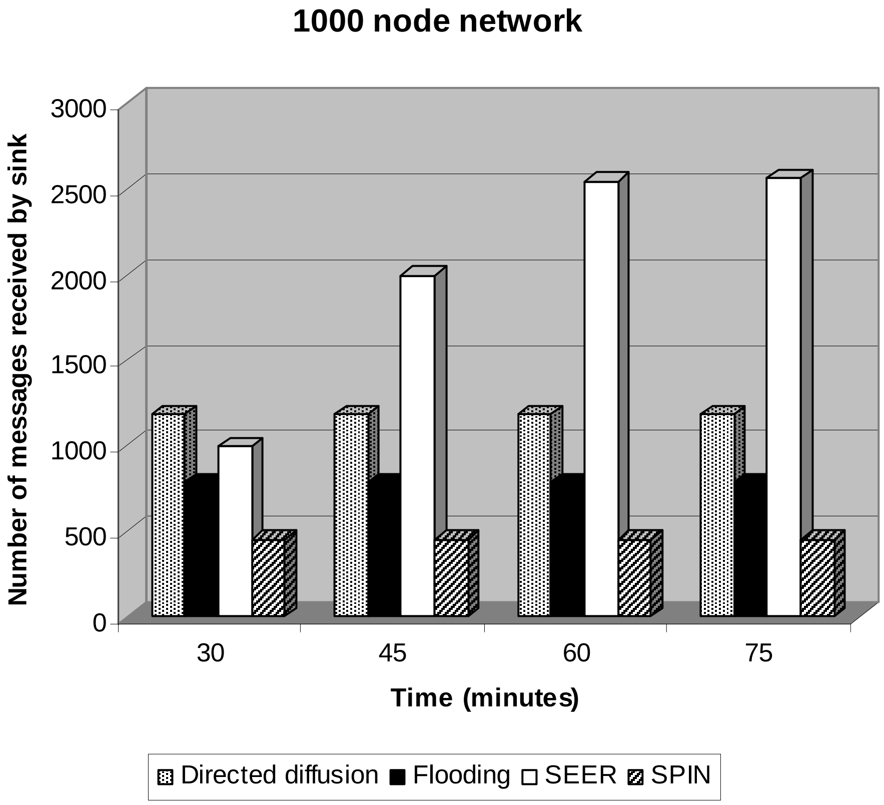

Figure A.19: The number of data messages received by the sink in a 1000 node network at selected intervals.

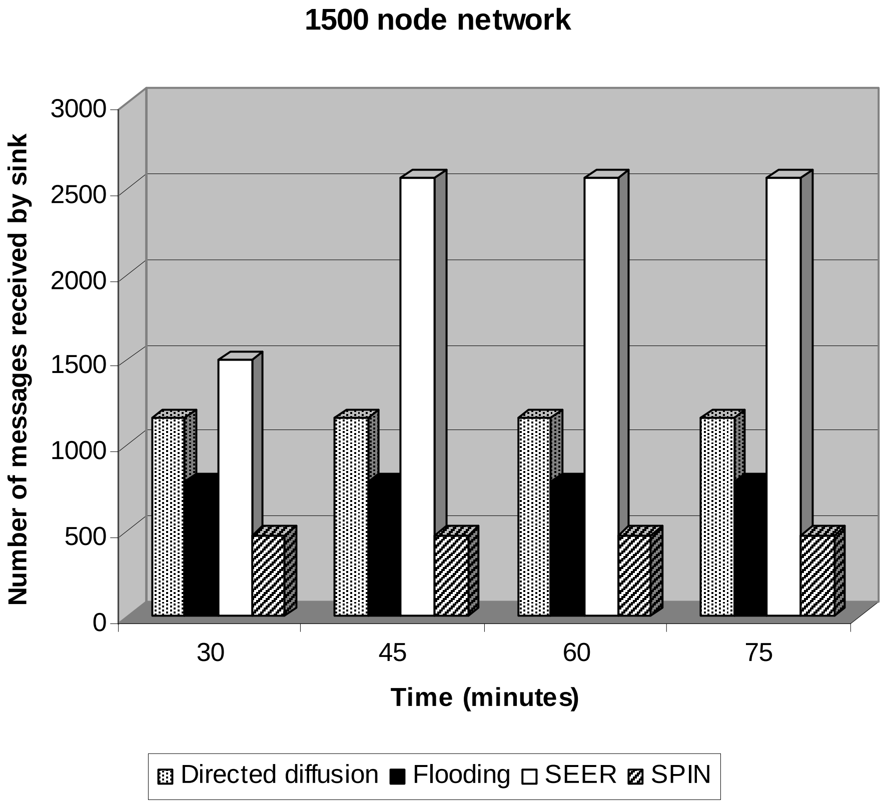

Figure A.20: The number of data messages received by the sink in a 1500 node network at selected intervals.