Mälardalen University Licentiate Thesis
No.79

Centralized Routing for Prolonged Network Lifetime in Wireless Sensor Networks

Ewa Hansen

January 2008

MÄLARDALEN UNIVERSITY

School of Innovation, Design and Engineering
Mälardalen University
Västerås, Sweden

ISSN 1651-9256
ISBN 978-91-85485-69-7
Printed by Arkitektkopia, Västerås, Sweden
Distribution: Mälardalen University Press

Abstract

In this thesis centralized routing methods for wireless sensor networks have been studied. The aim has been to prolong network lifetime by reducing the energy consumed by sensor-node communication.

Wireless sensor networks are rapidly becoming common in application areas where information from many sensors is to be collected and acted upon. The use of wireless sensor networks adds flexibility to the network, and the cost of cabling can be avoided.

Wireless sensor networks may consist of hundreds or even up to thousands of small compact devices, equipped with sensors (e.g. acoustic, seismic or image), that form a wireless network. Each sensor node in the network collects information from its surroundings and sends it to a base station, either from sensor node to sensor node, i.e. multihop, or directly to the base station i.e., singlehop.

We have made simulations that show that asymmetric communication with multihop extends the lifetime of large wireless sensor networks. We have also investigated the usefulness of enforcing a minimum separation distance between cluster heads in a cluster based wireless sensor network. The results show that our wireless sensor network performs up to 150% better when introducing a minimum separation distance between cluster heads. The simulations also show that the minimum separation distance resulting in the lowest energy consumption in our network varies with the number of clusters. Furthermore we have made an initial study of maximum lifetime routing in sparse wireless sensor networks to be able to see how different heuristic routing algorithms influence the energy consumption of individual sensor nodes, and thus the lifetime of a sparse sensor network. We have compared the maximum lifetime of the heuristic algorithms to the maximum lifetime of an optimal routing solution. These simulations show that for some types of applications the choice of heuristic algorithm is more important to prolong network lifetime than for other types of applications.

Swedish Summary - Svensk Sammanfattning

I denna avhandling har centraliserade vägvalsmetoder för trådlösa sensornätverk studerats. Målet har varit att förlänga nätverkens livslängd genom att minska energiåtgången för sensornodernas kommunikation.

Trådlösa sensornätverk blir en allt vanligare tillämpning där information från många sensorer behöver samlas in och bearbetas. Användandet av trådlösa sensornätverk ökar nätverkets flexibilitet, och kostnader för kabeldragning kan undvikas.

Trådlösa sensornätverk kan bestå av hundratals eller ända upp till tusentals små enheter, utrustade med en eller flera sensorer (för t.ex. ljud, ljus, rörelse eller bild), som formar ett trådlöst nätverk. Varje sensornod i nätverket samlar information från sin omgivning som den sedan skickar till basstationen antingen från sensornod till sensornod, s.k. multihop, eller direkt till basstationen, s.k. singelhop.

Vi har gjort simuleringar som visar att asymmetrisk kommunikation tillsammans med multihop ökar livslängden för stora trådlösa sensornätverk. Vi har också undersökt användbarheten av att upprätthålla ett minimiavstånd mellan klusterhuvuden i ett klusterbaserat sensornätverk. Resultaten visar att vårt trådlösa sensornätverk presterar upp till 150% bättre när ett minimiavstånd mellan klusterhuvuden används, mätt i antalet mottagna meddelanden hos basstationen. Simuleringarna har också visat att det minimiavstånd mellan klusterhuvudena som genererar den lägsta energikonsumtionen för nätverket varierar med antalet kluster.

Vi har även gjort en första studie där vi studerat hur man kan välja väg genom nätverket för att maximera livslängden i ett glest sensornätverk. Studien har gjorts för att se hur olika heuristiska algoritmer påverkar energikonsumtionen för enskilda noder, och följaktligen också hela det trådlösa sen-

sornätverkets livslängd. Vi har också jämfört den maximala livstiden för de heuristiska algoritmerna med den maximala livstiden för en optimal lösning. Simuleringarna har visat att för vissa typer av tillämpningar är valet av heuristisk algoritm mer viktigt för nätverkets livslängd, än för andra typer av tillämpningar.

Till Marcus
Jag älskar dig! Du är mitt allt!

Acknowledgements

First of all I would like to thank my supervisors Professor Mats Björkman, Professor Mikael Nolin and Doctor Dag Nyström, without you this work would have been impossible for me. I would like to thank my closest colleges, Martin Ekström, Marcus Blom and Mikael Ekström, for sharing your knowledge with me, and of course for all the coffee breaks! Also a big thanks to two of my former colleges Jonas Neander and Andreas Johnsson, you have been a great support, it wouldn't have been the same without you guys! Other persons I would like to thank are my colleges at the department. Thank you for all the nice discussions and for all the laughs we shared during these years, especially Lariza R, Peter W, Jörgen L, Andreas H, Monica W and Harriet E.

I would like to take the opportunity to thank my family for their believe in me and all the love and support I have got during these years. Thank you Mom, Dad, Elin, Inge, Marcus, Jonas, Josephine and Julia. I also want to say thanks to Sonya, Stefan, Robin, Jerry and Lena for letting me be a part of you life.

To my lovely boyfriend, Jonas. Thank you for always being there for me and for always supporting me, I love you with all my heart!

Ewa Hansen
Västerås, December 20, 2007

Contents

II Included Papers 35

9 Paper C: A Study of Maximum Lifetime Routing in Sparse Sensor Networks 77

List of Publications

Publications Included in the Licentiate Thesis

Paper A: *Asymmetric Multihop Communication in Large Sensor Networks*, Jonas Neander, Ewa Hansen, Mikael Nolin, Mats Björkman, In proceedings of the International Symposium on Wireless Pervasive Computing 2006, ISWPC, Phuket, Thailand, January, 2006.

Paper B: *Energy-Efficient Cluster Formation for Large Sensor Networks using a Minimum Separation Distance*, Ewa Hansen, Jonas Neander, Mikael Nolin and Mats Björkman, In proceedings of the Fifth Annual Mediterranean Ad Hoc Networking Workshop 2006, MedHocNet, Lipari, Italy, June 2006.

Paper C: *A Study of Maximum Lifetime Routing in Sparse Sensor Networks*, Ewa Hansen, Mikael Nolin and Mats Björkman, To appear in proceedings of the International Workshop on Wireless Ad Hoc and Mesh Networks 2008, WAMN, Barcelona, Spain, March 2008.

Chapter 5 describes these papers and my individual contribution for each paper in more detail.

Other Publications by the Author

Conferences and Workshops

- *Prolonging Network Lifetime in Long Distance Sensor Networks using a TDMA Scheduler*, Jonas Neander, Ewa Hansen, Jukka Mäki-Turja, Mikael Nolin and Mats Björkman, In proceedings of the Fifth Annual Mediterranean Ad Hoc Networking Workshop 2006, MedHocNet, Lipari, Italy, June 2006.

- *Prolonging Network Lifetime in Long Distance Sensor Networks using a TDMA Scheduler*, Jonas Neander, Ewa Hansen, Jukka Mäki-Turja, Mikael Nolin, Mats Björkman, Real-Time in Sweden (RTiS), SNART (the Swedish National Real-Time Association), Västerås, Sweden, August, 2007 (same paper as presented at MedHocNet 2006).

Technical Reports

- *Efficient Cluster Formation for Sensor Networks*, Ewa Hansen, Jonas Neander, Mikael Nolin and Mats Björkman, MRTC report ISSN 1404-3041 ISRN MDH-MRTC-199/2006-1-SE, Mälardalen Real-Time Research Centre, Mälardalen University, March, 2006.

- *A TDMA scheduler for the AROS architecture*, Jonas Neander, Ewa Hansen, Jukka Mäki-Turja, Mikael Nolin, Mats Björkman, MRTC report ISSN 1404-3041 ISRN MDH-MRTC-198/2006-1-SE, Mälardalen Real-Time Research Centre, Mälardalen University, March, 2006.

- *An Asymmetric Network Architecture for Sensor Networks*, Jonas Neander, Ewa Hansen, Mikael Nolin, Mats Björkman, MRTC report ISSN 1404-3041 ISRN MDH-MRTC-181/2005-1-SE, Mälardalen Real-Time Research Centre, Mälardalen University, August, 2005.

I

Thesis

Chapter 1

Introduction

In this thesis we show that asymmetric communication between a base station and the sensor nodes extends the lifetime of large centralized wireless sensor networks. We also show that enforcing a minimum separation distance between cluster heads, in a cluster based wireless sensor network, prolongs network lifetime. Furthermore, we show that for some types of applications, the choice of heuristic algorithm is more important to prolong network lifetime, than for other types of applications.

Wireless sensor networks are rapidly becoming common in application areas where information from many sensors is to be collected and acted upon. The use of wireless sensor networks adds flexibility to the network, and the additional cost of installation of cables can be avoided.

Wireless sensor networks consist of many small compact devices, equipped with sensors (e.g. acoustic, seismic or image sensors), that form a wireless network. Each sensor node in the network collects information from its surroundings, and sends it to a base station, either from sensor node to sensor node i.e. multihop, or directly to a base station i.e. singlehop.

A wireless sensor network may consist of hundreds or up to thousands of sensor nodes and can be spread out as a mass or placed out one by one. The sensor nodes collaborate with each other over a wireless media to establish a sensing network, i.e. a wireless sensor network. Because of the potentially large scale of the wireless sensor networks, each individual sensor node must be small and of low cost. The availability of low cost sensor nodes has resulted in the development of many other potential application areas, e.g. to monitor large or hostile fields, forests, houses, lakes, oceans, and processes in industries. The sensor network can provide access to information by collecting, processing, analyzing and distributing data from the environment.

In many application areas the wireless sensor network must be able to operate for long periods of time, and the energy consumption of both individual sensor nodes and the sensor network as a whole is of primary importance. Thus energy consumption is an important issue for wireless sensor networks.

1.1 Assumptions in this Thesis

In this thesis we assume that all wireless sensor nodes are battery operated, without the possibility to be recharged once connected to the sensor network. We also assume that a base station has global knowledge about all sensor node positions and that all sensor nodes in the network are relatively static. We also assume that the base station has "unlimited" power supply and high calculation capacity.

The simulations presented are made in the AROS framework [12]. The main feature of AROS is asymmetric communication, where the base station reaches all sensor nodes in the local network, but the sensor nodes may have to use several hops to reach the base station, see figure 1.1. A centralized approach is used where the base station makes all decisions about e.g. routing and scheduling. To be able to make the sensor nodes go into a sleep mode between sending and/or receiving, the nodes are assumed to be time synchronized and they are also assumed to be scheduled to avoid collisions.

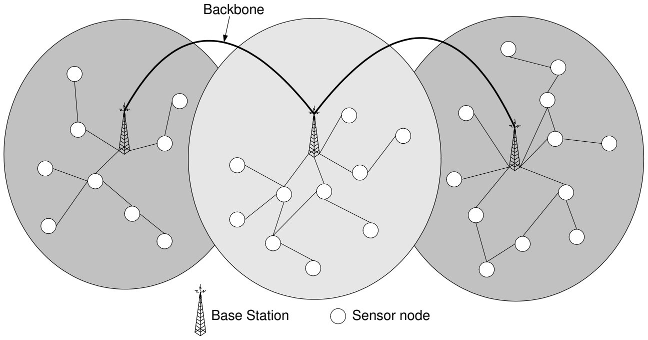

Figure 1.1: The AROS architecture

1.2 Method

The work presented in this thesis is based on simulations. Simulations allow for more aspects to be evaluated and more parameter values to be investigated, than what would be possible using real applications in real wireless sensor networks. Using simulations, we can evaluate a large number of alternative algorithms, and a large number of implementation strategies.

However, the accuracy of every simulation is dependent on the accuracy of the simulation models used. We therefore plan to implement the most promising of our algorithms and implementation strategies in real sensor networks in order to corroborate our simulation results.

1.3 Thesis Outline

Chapter 1: This chapter briefly introduces the wireless sensor network area.

Chapter 2: In this chapter we describe the wireless sensor network in more detail. We introduce some possible application areas where sensor networks are usable. We also give you a short overview of the design of the sensor node as well as network design issues.

Chapter 3: In this chapter we present the studied problem areas in wireless sensor networks.

Chapter 4: Related work is presented in this chapter.

Chapter 5: In this chapter we summarize and present the contributions of the papers included in this thesis.

Chapter 6: We conclude Part I with a conclusion and point out some directions for future work.

Chapter 2

Wireless Sensor Networks

After a short introduction to the sensor network area, in chapter 1, a more detailed presentation of the area will be presented in this chapter. For simplicity, the sensor networks throughout the thesis are assumed to be wireless unless otherwise explicitly stated.

2.1 Applications

As mentioned earlier, wireless sensor networks have many potential applications areas, e.g. military sensing, air traffic control, traffic observation, physical security, video surveillance, industrial and manufacturing automation, environment monitoring, building and structure monitoring, and hospital and health care monitoring [1, 7, 17, 18, 20].
Some of the application areas where sensor networks can be used are:

- Applications for military use: to detect and collect information about e.g. enemy movements, chemical-, biological-, nuclear attacks and materials.
- Applications for monitoring environmental changes in e.g. plains, forests, oceans, fields.
- Applications for monitoring vehicle traffic on highways to collect information about e.g. congested parts of a city.

Application areas more relevant for this thesis are areas where existing infrastructure can be used to support the sensor network are described below.

- Applications for industrial, use to monitor e.g. machines to get an increased knowledge about how the machine functions and about the production quality. For example, rolling machines at pulp and paper mills are big and complex. A sensor network can detect very small variations in speed and temperature that can have serious effects on the quality of the paper. A sensor network can also monitor the health of the staff working as well as the working environment e.g. temperature and ventilation.
- Applications for patient care both in and outside the hospital. For example, patients in hospitals that need some kind of health monitoring can use wireless sensor nodes instead of cabled sensor nodes and thus be more mobile.

  Another example is the continuous monitoring of patients or elderly outside the hospital to enable early detection of bad conditions and diseases for e.g. risk patients.

2.2 Sensor Node Design

Conceptually, a sensor node consists of a power unit, sensing unit, processing unit and radio unit that is able to both transmit and receive data (transceiver). Sometimes the sensor node also has a mobility unit as well as a localization unit, e.g., a global positioning system (GPS), see Figure 2.1.

Sensing

The sensing unit consists of two subunits, one or a group of sensors and an analog-to-digital converter (ADC). The ADC converts analog signals from the sensors to digital signals, used by the processing unit. The sensors are devices that respond to changes in the surroundings. The type of sensors being used on a sensor node depends on the application. The sensors can monitor speed, temperature, pressure, movement, humidity or vibrations to name a few.

Processing

The processing unit, usually a low speed CPU with small storage capabilities, performs tasks like routing and processing of sensed data etc. The choice of processing unit also determines, to a great deal, both the energy consumption as well as the computational capability of a sensor node.

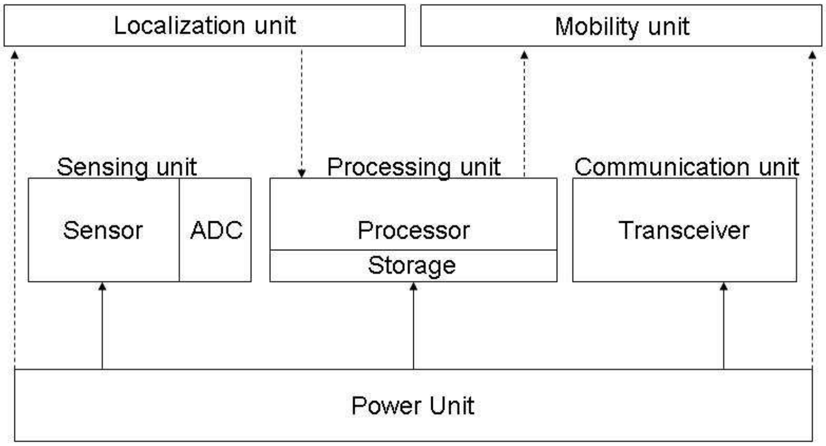

Figure 2.1: Architecture of a Sensor node

Communication

The transmission between sensor nodes is wireless and can be implemented by radio, infrared or other optical media. Much of the current hardware for sensor nodes is based on radio link communication.

Power

The power unit provides power to the other units and is typically a battery. Since the battery limits the amount of energy available to the node, this affects the lifetime of the node, thus in the end it also affects the lifetime of the sensor network. In many application scenarios, replacement or recharging (by e.g., solar cells or vibrations) of power resources is costly or even impossible.

The most power-consuming activity of a sensor node is typically communication [13]. Hence, communication must be kept to an absolute minimum in order to maximize the lifetime of the sensor nodes. All activities involving communication (sending, receiving, listening for data) are power-consuming and one important way to save power is to have the communicating device turned off as much as possible.

2.3 Network Design

The design of a sensor network is influenced by many factors, including fault tolerance, scalability, production cost, network topology, hardware constraints, transmission media, and power consumption [1].

2.3.1 Routing

Since a sensor network can cover a large area, conventional techniques such as sending information directly from each sensor node to a base station can result in long distance communication which in many cases needs to be avoided. To avoid problems with long distance communication, so called multihop communication can be used. Information is then sent from sensor node to sensor node to finally reach a base station, thus routing mechanisms/techniques are needed to send information between nodes in such a network.

In a sensor network with battery operated sensor nodes, the lifetime and the power consumption become very important, and many researchers are focusing on designing energy efficient routing protocols that prolong network lifetime. The design of energy efficient routing protocols that prolongs network lifetime is complex and to find optimal solution are known to be NP-hard[1] [2].

Symmetric and Asymmetric Communication

To decrease some of the complexity, a base station can make routing decisions instead of each individual node, a centralized approach. To distribute the information about routing for each node, symmetric or asymmetric communication can be used. When symmetric communication is used, the base station sends routing information with multihop until it reaches the end destination. The nodes use the same route when sending their sensed information to the base station. The symmetric approach will increase energy consumption of the nodes used for routing the information to and from the base station.

Using asymmetric communication can make the energy consumption more distributed among the sensor nodes. One way of asymmetric communication is to use multihop, but with different routes to and from the base station. The total energy consumption will be similar to the energy consumption when using symmetric communication but it will be distributed differently. Another asymmetric approach is to send information from the base station directly to

---

[1] Nondeterministic polynomial-time Hard.

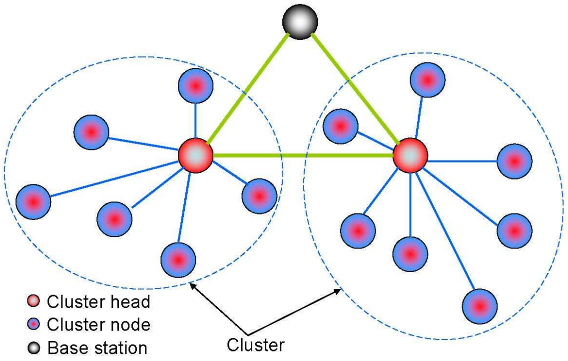

Figure 2.2: One kind of cluster hierarchy in a sensor network

each sensor node. This will decrease the total energy consumption of the sensor nodes as well as the individual energy consumption of the sensor nodes that otherwise would have been involved in the distribution of information from the base station.

2.3.2 Data Aggregation

To reduce the amount of traffic in the network, hence saving energy, we can often aggregate, or fuse, data. When aggregating or fusing data, the amount of data forwarded is reduced by processing of the data in each forwarding node. One way of aggregation is to remove all redundant information. For example, if several nodes send the same information, the forwarding node can forward one packet instead of several packets with the same information, thus reducing traffic and saving energy. Another way is to process data, by summarizing or computing a mean value of e.g. temperature, and then forward this to the base station. This is often called fusion.

2.3.3 Clustering

Clustering is one way of making routing less complex, and for some sensor networks, more energy efficient. In clustering, adherent cluster nodes send their data to a central cluster head, and the cluster head then forwards this data towards a base station, see Figure 2.2.

However, one drawback with clustering is that the cluster head will use more energy than non cluster head nodes, when listening for or receiving information/data. The cluster heads may also send data long distances to reach the base station or another cluster head, thereby using a lot of energy. To avoid draining the energy of these cluster heads, the selection of cluster head needs to be changed several times during the lifetime of a sensor network [3, 11].

To decrease routing complexity and increase energy efficiency it is important to decide how many cluster heads that are most suitable, and which of the sensor nodes are going to act as cluster heads.

Large numbers of sensing nodes may congest the network with information. To solve this problem, some sensors, such as the cluster heads, can aggregate data, and then send the new information towards the base station.

Chapter 3

Studied Problem Areas

Compared to traditional networks, sensor networks have rather different characteristics and quality measurements. Because of the high collaboration of sensor nodes and very specific application goals, there is no "one size fits all" solution to routing, so the specific characteristics decide what routing mechanism to use.

In this thesis we have made simulations that show that asymmetric communication with multihop extends the lifetime of large cluster based sensor networks. We have also investigated the usefulness of enforcing a minimum separation distance between cluster heads in a cluster based sensor network to prolong network lifetime.

3.1 Multihop Communication

As mentioned in chapter 2, a large number of sensor nodes have to work together and techniques such as sending information directly from each sensor node to a base station need in many cases to be avoided. When a sensor node sends data directly to a base station, the amount of energy used by the sensor node can be quite high, depending on the location of the sensor node relative to the base station. In such a scenario, the nodes that are furthest away from the base station will run out of power much faster than those nodes that are closer to the base station, and parts of the network area will no longer be covered by functional sensor nodes. When communicating in a sensor network the amount of energy used by a sensor node depends on e.g. the size of the packet and the communication distance. The amount of energy used when communicating can be proportional to up to d^4 (d = distance between the two communicating nodes), for long distance communication [4]. To avoid problems with long distance communication, so called multihop communication is used. In multihop, information is sent from sensor node to sensor node to finally reach the base station, thus routing mechanisms/techniques are needed.

We have, in paper A, made simulations that show that multihop communication together with asymmetric communication between the base station and the sensor nodes are less energy consuming than not using asymmetric communication.

The simulations are made in the AROS architecture [12] where the base station acts as a master for the sensor nodes and is able to reach all its sensor nodes in one hop. However, all sensor nodes might not reach the base station in one hop, hence other nodes might need to forward information towards the base station, i.e. multihop. In the AROS architecture we use cluster heads to forward information.

3.2 Cluster Head Selection

Clustering is one way of making routing less complex, and for some sensor networks, more energy efficient.

To decrease routing complexity and increase energy efficiency it is important to decide how many cluster heads that are most suitable, and which of the sensor nodes are going to act as cluster heads. Another important issue is the geographical placement of the cluster heads. If the cluster heads are grouped together or located too close to each other, the adherent cluster nodes need to communicate very long distances and thereby draining their energy. The size of the clusters are also likely to vary, some clusters may be very small and others very large (many nodes belong to one cluster head).

To be able to know that the cluster heads are not too close to each other, we have in paper B made simulations to investigate the usefulness of enforcing a minimum separation distance between cluster heads in a cluster based sensor network. The simulations, made in the AROS architecture, indicates that enforcing a minimum separation distance increases network lifetime and that the number of clusters used also influences the lifetime of the network.

3.3 Heuristic Routing Algorithms

As mentioned in chapter 2, the most power-consuming activity of a sensor node is communication. Hence, communication cost must be as small as possible in order to save power.

One approach to minimize energy consumption is to always use the route that is least energy expensive to reach the base station. But if all traffic is routed through the minimum energy path (the least energy expensive way), the sensor nodes in this path will drain their energy and the network lifetime will be affected. To avoid this problem, routing paths will have to be changed several times during the lifetime of the network, and the energy consumption need to be balanced among the senor nodes to maximize the network lifetime.

In paper C, an initial study of maximum lifetime routing in sparse sensor networks has been made to be able to see how different heuristic routing algorithms influence the energy consumption for individual sensor nodes, and thus the lifetime of a sparse sensor network. The maximum lifetime of the heuristic algorithms is also compared to the maximum lifetime of an optimal routing solution.

Chapter 4

Related Work

In this chapter we present some related work. We begin with some network architectures related to the AROS architecture, and thereafter some routing methods for prolonged lifetime in sensor networks are presented.

4.1 Wireless Sensor Network Architectures

In this section some of the related work to the AROS architecture is described.

4.1.1 The LEACH project

LEACH (Low-Energy Adaptive Clustering Hierarchy) [4] is a well known cluster based architecture where a node elects itself to be cluster head, by some probability, and broadcasts an advertisement message to all the other nodes in the network. A non-cluster head node selects a cluster head to join based on the received signal strength. All nodes in the network have the potential to be cluster head during some periods of time. A TDMA[1] scheme starts every round with a set-up phase. The next phases consist of several cycles where all nodes have their slots periodically. The nodes send their data to the cluster head that aggregates the data and send it to its base station at the end of each cycle. After a certain amount of time, the round ends and the network reenter the set-up phase.

---

[1] Time Division Multiple Access.

LEACH-C (LEACH-Centralized) [3] has been developed out of LEACH and the basis for LEACH-C is to use a central control algorithm to form clusters. The base station runs the centralized cluster formation algorithm to determine the clusters for that round. To determine clusters and select cluster heads, LEACH-C uses simulated annealing [10] to search for near-optimal clusters.

A further development is LEACH-F (LEACH with Fixed clusters) [3]. LEACH-F is based on clusters that are formed once - and then fixed. The cluster head position then rotates among the sensor nodes in the cluster.

The main drawback with the LEACH protocols is that all the sensor nodes communicate directly with the base station, so called symmetric singlehop communication. When the network size increases, the communication distance will be long, thus draining some of the sensor nodes of power very quickly. If using asymmetric communication with multihop communication from the sensor nodes to the base station, as in the AROS architecture, the energy will, for the majority of the nodes, last longer (since shorter communication distances are used).

4.1.2 PEGASIS

PEGASIS (Power-Efficient GAthering in Sensor Information System) [6], a near optimal chain-based protocol. PEGASIS avoids cluster formation and uses only one node in a chain to transmit to a base station, instead of multiple nodes. The key idea in PEGASIS is to form a chain among the sensor nodes so that each node will receive from and transmit to a close neighbor. Gathered data moves from node to node, gets fused, and then, eventually, an elected node transmits the data to a base station.

4.1.3 TEEN and APTEEN

TEEN (Threshold-sensitive Energy Efficient sensor Network protocol) [8] and APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol) [9] are both designed for time-critical applications. Both TEEN and APTEEN uses asymmetric communication between the base station and the sensor nodes. Further, they build clusters with cluster heads that perform data aggregation and then send the aggregated data to the base station or to a cluster head.

In TEEN, the cluster head broadcasts a hard and a soft threshold to its members. The hard threshold aims at reducing the number of transmissions by allowing the nodes to transmit only when the sensed attribute is in the range of interest. The soft threshold further reduces the number of transmissions by eliminating all the transmissions which might have occurred otherwise when there is little or no change in the sensed attribute. The soft threshold can be varied, depending on how critical the sensed attribute and the target application are.

APTEEN is a hybrid protocol that changes the periodicity or threshold values used in the TEEN protocol according to the user needs and the type of the application. In APTEEN, the cluster head broadcasts physical parameter attributes important for the user. APTEEN sends periodic data to give the user a complete picture of the network. APTEEN also responds immediately to drastic changes for time-critical situations.

Both TEEN and APTEEN are designed to reduce the amount of messages in the network, hence, prolong the lifetime of the network. TEEN and APTEEN send data after a certain threshold which will result in longer delay times and thereby prolonged network lifetime.

AROS as well as these two protocols use asymmetric communication and they are designed to prolong the network lifetime. One drawback is however that the cluster heads in APTEEN broadcast e.g. the threshold values to the senor nodes, which is energy consuming. In the AROS architecture, the base station does all the communication to the sensor nodes. Another benefit of using AROS is that the cost per packet is low which results in long network lifetime. Using threshold values in the AROS architecture as in TEEN could prolong network lifetime even more.

4.1.4 BCDCP

BCDCP (Base-station Controlled Dynamic Clustering Protocol) [11] is a centralized routing protocol with a high energy base station that makes all the highly energy consuming activities, e.g. selecting cluster heads and routing paths, and performing randomized rotation of cluster heads. The idea in BCDCP is to organize balanced clusters with uniform placement of cluster heads where each cluster head serves an approximately equal number of member nodes.

During each setup phase the base station receives information on the current energy status from all the nodes in the network. BCDCP uses an iterative splitting algorithm to form clusters. The first step is to choose two nodes, among the eligible nodes, that have the maximum separation distance. Step two is to group the remaining nodes to one of the cluster heads, whichever is closest. Step tree is to balance the clusters so that each cluster has approximately the same number of nodes. Step four is to start from step one and split the sub-clusters in to smaller parts. The iteration of the four steps continues until the desired number of cluster heads is attained.

BCDCP is one of the inspiration sources to paper B.

4.2 Routing Algorithms for Wireless Sensor Networks

Many researchers have focused on energy efficient routing and power aware routing, e.g. [4, 11, 16, 19] to name a few.

One of the early power saving protocols was proposed by Singh *et* al. in [15] where they presented the PAMAS protocol. The PAMAS protocol is a MAC[2] layer protocol that turns off the radio when the node is not transmitting or cannot receive packets. This protocol saves 40-70% of battery power according to [15]. The paper also includes several power aware metrics that are used to construct energy efficient routes e.g. *Minimize Energy consumed/packet* and *Maximize Time to Network Partition*.

In [5] Li *et* al. presents the *max-min* $z\mathrm{P}_{min}$ algorithm. The *max-min* $z\mathrm{P}_{min}$ algorithm combines the benefit of selecting path with both the minimum power consumption and the path that maximizes the minimal remaining power in the nodes of the network. An important factor in the *max-min* $z\mathrm{P}_{min}$ algorithm is the parameter z that tries to find a balance between the maximum minimum residual power path and the minimal power consumption path, but it seems that it is not so easy to find the optimal value of z. According to [5], the algorithm requires knowledge about each node in the network which can be a problem when implementing the algorithm in large networks. To solve this problem they propose a zone-based routing that relies on *max-min* $z\mathrm{P}_{min}$ but is scalable. In zone-based routing the network is divided into smaller zones, and each zone has only control over how to route the messages within its own zone. A global path across zones is also computed.

Chang *et* al. in [2] presents a Flow Augmentation algorithm (FA) which is a shortest cost path routing where the link cost is a combination of transmission and reception energy consumption and the residual energy level at the two end nodes. The objective in [2] is to find the best link cost function which leads to the maximization of the network lifetime. When there is plenty of remaining energy in the nodes, the energy cost term is emphasized, but when the node has less remaining energy, the remaining energy term has greater impact, i.e. is

---

[2] Media Access Control.

given more weight in the cost function.

In [14], Shah *et* al. proposes a scheme, called Energy Aware Routing, that uses sub-optimal communication paths occasionally. The basic idea behind the scheme is to increase the survivability of the network by sometimes communicating through a sub-optimal path. They use a set of good paths and choose one of them, based on some probabilistic function. This means that instead of using one single communication path, different communication paths will be chosen at different times, thus any single communication path will not suffer from energy exhaustion.

Since sensor networks have very different specific application goals, there is no "one size fits all" solution, and new power and energy efficient routing techniques are needed. Our work in paper C is a first attempt to map this area and to find the relevant tradeoffs.

Chapter 5

Summary of papers and their Contributions

5.1 Paper A: Asymmetric Multihop Communication in Large Sensor Networks

Asymmetric Multihop Communication in Large Sensor Networks, Jonas Neander, Ewa Hansen, Mikael Nolin, Mats Björkman, In proceedings of the International Symposium on Wireless Pervasive Computing 2006, ISWPC, Phuket, Thailand, January, 2006.

In this paper we presented a simulation comparison between asymmetric and symmetric communication. We did this by comparing LEACH [4], which uses symmetric communication, to a new extension of LEACH called AROS, Asymmetric communication and ROuting in Sensor networks.

The main focus of the comparisons was to study the energy consumption when transferring data from the sensor nodes to the base station. This comparison was done to verify that, in large networks, forwarding data is more energy efficient than sending it directly to the base station. In this paper we showed that LEACH with the new extension AROS delivers more messages to the base station than before, given the same amount of energy. We also showed that AROS has more sensor nodes alive at any given time, after the first demised sensor node.

This is a joint paper and I have together with my co-worker and co-writer

Jonas Neander implemented AROS and performed the simulations in NS-2.

5.2 Paper B: Energy-Efficient Cluster Formation for Large Sensor Networks using a Minimum Separation Distance

Energy-Efficient Cluster Formation for Large Sensor Networks using a Minimum Separation Distance, Ewa Hansen, Jonas Neander, Mikael Nolin and Mats Björkman, In proceedings of the Fifth Annual Mediterranean Ad Hoc Networking Workshop 2006, MedHocNet, Lipari, Italy, June 2006.

In this paper we made simulations to investigate the usefulness of enforcing a minimum separation distance between cluster heads in a cluster based sensor network. The idea is to prolong network lifetime by spreading the cluster heads, thus lowering the average communication energy consumption.

We showed that using a minimum separation distance between cluster heads improves energy efficiency, measured by the number of messages received at the base station. We also showed that it is better, up to 150% in our simulations, to use a minimum separation distance between cluster heads than not to use any minimum separation distance. By using a minimum separation distance between cluster heads we make the network live longer, gathering data from the whole network area. We also showed that the number of clusters used together with the minimum separation distance affects the energy consumption.

Our simulations also showed that, depending on the number of dead nodes that can be tolerated, different minimum separation distances as well as different number of clusters affect the number of messages received before the given tolerance limit is reached.

I performed most of the work behind this paper and I was the main driving author and I wrote most of the text for the paper.

5.3 Paper C: A Study of Maximum Lifetime Routing in Sparse Sensor Networks

A Study of Maximum Lifetime Routing in Sparse Sensor Networks, Ewa Hansen, Mikael Nolin and Mats Björkman, To appear in proceedings of the International Workshop on Wireless Ad Hoc and Mesh Networks 2008, WAMN, Barcelona, Spain, March 2008.

In this paper we presented an initial study of maximum lifetime routing in sparse sensor networks. We have studied simulations of how different heuristic routing algorithms influence the energy consumption of individual sensor nodes, and thus the functional lifetime of a sparse sensor network. We have also compared the maximum lifetime of the heuristic algorithms to the maximum lifetime of an optimal routing solution.

We have performed simulations with 100 randomly generated sensor networks where the network area was 400x400 m^2 and the number of nodes randomly spread across the network was 5. The simulations were made with both aggregation and non-aggregation of data, and a comparison with an optimal routing solution was also done.

The conclusions of these simulations were that when aggregating data, the choice of heuristic algorithm was not as significant as when not aggregating data. Our simulations with non-aggregated data indicated that using only one of the presented heuristic routing algorithms is not enough to find a near optimal routing.

I performed most of the work behind this paper and I was the main driving author and I wrote most of the text for the paper.

Chapter 6

Conclusions and Future Work

In this thesis we have presented a simulation comparison between asymmetric and symmetric communication in sensor networks. The simulations were made in the AROS architecture.

The simulations in paper A showed that AROS has 25% of its energy left when the LEACH protocols have used their energy and demised. The simulations also showed that asymmetric communication with multihop extends the lifetime of the sensor nodes in large sensor networks.

We have also performed simulations in order to determine how much we can lower the energy consumption in the sensor network by separating the cluster heads, i.e., by distributing the cluster heads through the whole network. In paper B, we presented a simple energy-efficient cluster formation algorithm for the AROS architecture. The simulations showed that using a minimum separation distance between cluster heads improves energy efficiency up to 150% compared to not using a minimum separation distance, measured by the number of messages received at the base station. By using a minimum separation distance between cluster heads we can make the network live longer, gathering information from the whole network area.

In paper C, an initial study of maximum lifetime routing in sparse sensor networks is made, to see how different heuristic routing algorithms influence the energy consumption for individual sensor nodes, and thus the lifetime of a sparse sensor network. We have also compared the maximum lifetime of the heuristic algorithms to the maximum lifetime of an optimal routing solution.

In this simulation study we have used both aggregation of data as well as non-aggregation of data when forwarding.

When aggregating data the differences are not very big among the heuristic algorithms. Comparing results from the optimal routing solution to results from the heuristic algorithms, the differences are very small or none.

When not aggregating data when forwarding, the differences among the heuristic algorithms were slightly bigger. Comparing results from the heuristic algorithms to results from the optimal routing solution, the differences where more significant, when comparing the total number of rounds. None of the heuristic algorithms could match the optimal solution. The results of these simulations are that when aggregating data, the choice of heuristic algorithm is not as significant as when not aggregating data. In other words, for some types of applications the choice of heuristic algorithm is more important, than for other types of applications.

Future Work

In this thesis we have shown that we can prolong network lifetime by making intelligent routing decisions. Our simulations have indicated that with non-aggregated data, using one of the presented heuristic routing algorithms is not enough to find a near optimal routing, hence it is possible that several different heuristic algorithms need to be combined to find a near optimal routing solution.

In the future we will continue our work to prolong network lifetime, e.g. until the first node demises (in sparse networks) or until some threshold of nodes have demised (in more densely populated networks).

Our aim is to find a near optimal routing solution by e.g. weighting each link so that no node drains its energy faster than the other nodes, i.e. avoiding hotspots. After evaluating what algorithms that are most suitable, real world experiments will be done. This in order to verify that our simulations can be used as an approximation of the reality.

We want to study networks with more dense nodes and evaluate different heuristic algorithms that prolongs network lifetime. We also want to evaluate the use of clustering in a centralized sensor network. We believe that as we have global knowledge about the sensor network, the base station can make intelligent routing decisions and clustering may not be the most energy efficient technique.

Bibliography

[1] I. F. Akyildiz, Su Weilian, Y. Sankarasubramaniam, and E. E. Cayirci. A Survey on Sensor Networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[2] J. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(4):609–619, 2004.

[3] W. Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts institute of technology, June 2000.

[4] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Maui, Hawaii, Jan 2000. In Proceedings of the 33rd International Conference on System Sciences (HICSS '00).

[5] Q. Li, J. Aslam, and D. Rus. Online power-aware routing in wireless Adhoc networks. In Proceedings of the 7th annual international conference on Mobile computing and networking, 2001.

[6] S. Lindsey and C. S. Raghavendra. PEGASIS: Power-Efficient GAthering in Sensor Information Systems. volume 3, pages 1125–1130. Aerospace Conference Proceedings, 2002. IEEE, March 2002.

[7] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Andersson. Wireless Sensor Networks for Habitat Monitoring. *WSNA'02*, September 2002.

[8] A. Manjeshwar and D. P. Agrawal. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. *Parallel and Distributed Processing Symposium., Proceedings 15th International*, pages 2009–2015, April 2001.

[9] A. Manjeshwar and D. P. Agrawal. APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks. *Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002*, pages 195–202, April 2002.

[10] T. Maruta and H. Ishibuchi. Performance Evaluation of Genetic Algorithms for Flowshop Scheduling Problems. *Proceedings of the 1st IEEE Conference on Evolutionary computation*, 2:812–817, June 1994.

[11] S. D. Muraganathan, D. C. F. Ma, R. I Bhasin, and A. O. Fapojuwo. A centralized energy-efficient routing protocol for wireless sensor networks. *Communications Magazine, IEEE*, 43(3):8–13, March 2005.

[12] Jonas Neander. Using existing infrastructure as support for wireless sensor networks. Licentiate thesis, June 2006.

[13] G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[14] R. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. *In Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, March 2002.

[15] S. Singh, M. Woo, and C. S. Raghavendra. Power-Aware Routing in Mobile Ad Hoc Networks. In *Mobile Computing and Networking*, pages 181–190, 1998.

[16] S. Ming Tseng and R. Pandey. A Hierarchical Routing Protocol for Networks of Heterogeneous Sensors. Goteborg, August 2004. 10th International Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA).

[17] M. Tubaishat and S. Madria. Sensor Networks: An Overview. *IEEE Potentials*, pages 20–23, April/May 2003.

[18] L. Yu; N. Wang; X-Meng. Real-time forest fire detection with wireless sensor networks. International Conference on Wireless Communications, Networking and Mobile Computing.

[19] M. Younis, M. Youssef, and K. Arisha. Energy-aware management for cluster-based sensor networks. *Computer Networks*, 43:649–668, Dec 2003.

[20] M. Youssef, A. Yousif, N. El-Sheimy, and A. Noureldin. A Novel Earthquake Warning System Based on Virtual MIMO-Wireless Sensor Networks. Canadian Conference on Electrical and Computer Engineering, April 2007.

II

Included Papers

Chapter 7

Paper A: Asymmetric Multihop Communication in Large Sensor Networks

Jonas Neander, Ewa Hansen, Mikael Nolin and Mats Björkman
In International Symposium on Wireless Pervasive Computing 2006, ISWPC,
Phuket, Thailand, January, 2006

Abstract

With the growing interest in wireless sensor networks, energy efficient communication infrastructures for such networks are becoming increasingly important. In this paper, we compare and simulate asymmetric and symmetric communication in sensor networks. We do this by extending LEACH, a well-known TDMA cluster-based sensor network architecture, to use asymmetric communication. The extension makes it possible to scale up the network size beyond what is feasible with LEACH and its variants LEACH-C and LEACH-F.

7.1 Introduction

In this paper we present a simulation comparison between asymmetric and symmetric communication. We do this by comparing LEACH [1], which uses symmetric communication, to a new extension of LEACH called AROS, Asymmetric communication and ROuting in Sensor networks. We show that asymmetric multihop communication prolongs the lifetime of the sensor nodes in large networks. AROS is based on LEACH-C and LEACH-F [2] but uses the possibility to use asymmetric communication and forwarding of packets [3, 4].

With the growing interest in sensor networks, efficient communication infrastructures for such networks are becoming increasingly important. Among the interesting application areas for sensor networks are environmental surveillance and surveillance of equipment and/or persons in, e.g., factories or hospitals. Common for application areas considered in this paper are that sensor nodes are typically left unattended after deployment, the communication is wireless, and the power supply is limited.

Deploying unattended sensor nodes with limited power supplies implies that one important feature of a sensor network is its robust functionality in face of failing network nodes. Another implication is that, if the network is to survive a longer period of time, new nodes will have to be added to the existing network. Thus the network topology must be dynamically adaptable.

In AROS we use a semi-centralized approach where resource-adequate infrastructure nodes can act as base stations and, hence, be used to off-load sensors and thus prolong network lifetime. Often, the base stations can be situated in existing infrastructures. For instance, there are infrastructure networks built in hospitals and industrial factories that could be used to host base stations and thereby prolong the lifetime of the sensor networks. The infrastructure network can act as a, possibly fault tolerant, base station backbone for sensor nodes.

Industrial and hospital infrastructure networks are relatively static and they do not have limited energy as sensor nodes do. In this paper we assume that the base stations are stationary. The infrastructure network could be wired, wireless or a combination of both, see Figure 7.1.

A base station in LEACH-C, LEACH-F and AROS has large radio coverage and has the potential to accept all the sensor nodes that are receiving the signal from the base station. For some sensor nodes, it may be highly energy-consuming to communicate directly with a base station. The traffic from these sensor nodes should rather be forwarded by other sensor nodes in order to save energy.

One possible solution in order to reduce the amount of traffic in the net-

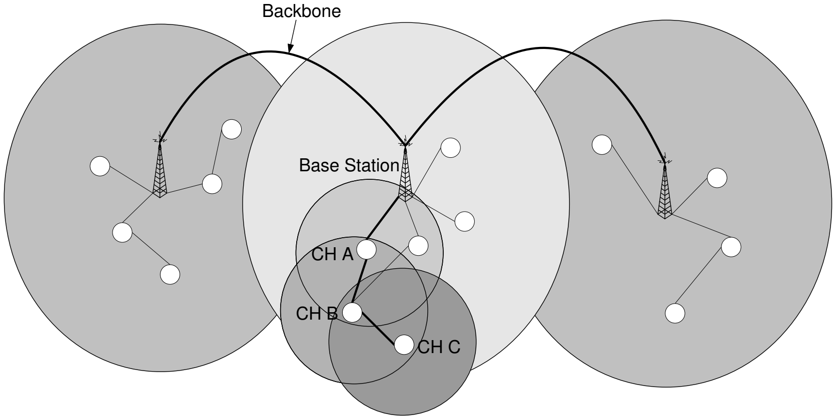

Figure 7.1: Overview of the architecture.

work is to build clusters of sensor nodes as proposed in e.g. [5, 1, 6]. Some sensor nodes become cluster heads and collect all traffic from/to their cluster. A cluster head aggregates the collected data and then sends it to its base station. In AROS, asymmetric communication is possible. That is, the base station reaches all the sensor nodes directly, while some sensor nodes cannot reach the base station directly but need other nodes to forward its data, hence routing schemes are necessary. Routing of traffic through other sensor nodes will increase the power consumption of the forwarding sensor nodes. Therefore, routing decisions must be carefully evaluated in order to maximize network lifetime. AROS extends LEACH-C and LEACH-F with multihop forwarding for traffic directed towards the base station.

The most power-consuming activity of a sensor node is typically radio communication [7]. Hence, communication must be kept to an absolute minimum. All activities involving communication are power-consuming and the most important way to save power is to turn off the radio as long time as possible. This applies to transmission and reception, but also to listening for data. Hence, as in LEACH and its variants LEACH-C and LEACH-F, we use Time Division Multiple Access (TDMA) schemes for sensor node communication. Using TDMA allows the radio to be turned off for long periods of time. AROS differs from LEACH and its variants when it comes to the cluster heads sending data to the base station. For this part of the communication, LEACH and its variants use CSMA while AROS uses TDMA.

In this paper we provide an initial simulation study comparing asymmetric multihop communications (AROS) and symmetric single hop communications, represented by the LEACH variants LEACH-C and LEACH-F. The main focus of the comparisons is to study the energy consumption when transferring data from the sensor nodes to the base station. We do these comparisons in order to verify that, in large networks, forwarding data is more energy efficient than sending it directly to the base station.

We show that LEACH with the new extension AROS delivers more messages to the base station than before, given the same amount of energy. We also show that AROS has more sensor nodes alive at any given time, after the first demised sensor node. Furthermore, the sensor nodes that are alive can be found throughout the entire network thus providing coverage of the whole monitored area. Our results show that AROS improves communication energy efficiency when the network size increases.

The rest of this paper is outlined as follows: in Section 7.2, we describe related work. In Section 7.3, the AROS architecture is presented. Section 7.4 describes the comparisons between AROS and the LEACH protocols, and Section 7.5 presents the results from the comparisons. Finally, we conclude and outline future work.

7.2 Related Work

LEACH (Low-Energy Adaptive Clustering Hierarchy) [1] is a TDMA cluster based approach where a node elects itself to be cluster head by some probability and broadcasts an advertisement message to all the other nodes in the network. A non-cluster head node selects a cluster head to join based on the received signal strength. Being cluster head is more energy consuming than to be a non-cluster head node, since the cluster head needs to receive data from all cluster members in its cluster and then send the data to the base station. All nodes in the network have the potential to be cluster head during some periods of time. The TDMA scheme starts every round with a set-up phase to organize the clusters. After the set-up phase, the system is in a steady-state phase for a certain amount of time. The steady-state phases consist of several cycles where all nodes have their slots periodically. The nodes send their data to the cluster head that aggregates the data and send it to its base station at the end of each cycle. After a certain amount of time, the TDMA round ends and the network re-enters the set-up phase.

LEACH-C (LEACH-Centralized) [2] has been developed out of LEACH and the basis for LEACH-C is to use a central control algorithm to form clusters. The protocol uses the same steady-state protocol as LEACH. During the set-up phase, the base station receives information from each node about their current location and energy level. According to [2], the nodes may get their current location by using a global positioning system (GPS) receiver that is activated at the beginning of each round. After that, the base station runs the centralized cluster formation algorithm to determine the clusters for that round. To determine clusters and select cluster heads, LEACH-C uses simulated annealing [8] to search for near-optimal clusters. Before running the algorithm that determines and selects the clusters, the base station makes sure that only nodes with “enough” energy are participating in the cluster head selection. Once the clusters are created, the base station broadcasts the information to all the nodes in the network. Each of the nodes, except the cluster head, determines its TDMA slot used for data transmission. Then, the node goes to sleep until it is time to transmit data to its cluster head.

A further development is LEACH-F (LEACH with Fixed clusters) [2]. LEACH-F is based on clusters that are formed once - and then fixed. Then, the cluster head position rotates among the nodes within the cluster. The advantage with this is that, once the clusters are formed, there is no set-up overhead at the beginning of each round. To decide clusters, LEACH-F uses the same centralized cluster formation algorithm as LEACH-C. The fixed clusters in LEACH-F do not allow new nodes to be added to the system and do not adjust their behavior based on nodes dying. Furthermore, LEACH-F does not handle node mobility.

TEEN (Threshold-sensitive Energy Efficient sensor Network protocol) [9] and APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol) [10] are both designed for time-critical applications. Both TEEN and APTEEN uses asymmetric communication between the base station and the sensor nodes. Further, they build clusters with cluster heads that perform data aggregation and then send the aggregated data to the base station or to a cluster head.

In TEEN, the cluster head broadcasts a hard and a soft threshold to its members. The hard threshold aims at reducing the number of transmissions by allowing the nodes to transmit only when the sensed attribute is in the range of interest. The soft threshold further reduces the number of transmissions by eliminating all the transmissions which might have occurred otherwise when there is little or no change in the sensed attribute. The soft threshold can be varied, depending on how critical the sensed attribute and the target application are.

APTEEN is a hybrid protocol that changes the periodicity or threshold values used in the TEEN protocol according to the user needs and the type of the application. In APTEEN, the cluster head broadcasts physical parameter attributes important for the user. APTEEN sends periodic data to give the user a complete picture of the network. APTEEN also responds immediately to drastic changes for time-critical situations.

Both TEEN and APTEEN are modified to reduce the amount of messages in the network, hence, increasing the lifetime of the network. However, a comparison between TEEN and APTEEN with LEACH and its variants, as in [9] and [10], is not directly suitable. LEACH sends data periodically to the base station while TEEN and APTEEN only send data after a certain threshold. This will result in longer delay times and prolonged network lifetime. LEACH and LEACH-C delivers more data than TEEN and APTEEN to the base station. Hence, LEACH and LEACH-C consume less energy per message than TEEN and APTEEN. Since TEEN and APTEEN are protocols for longevity only and do not consider the data throughput to the base station, it is beyond the scope of this paper to compare them with AROS. It is more suitable to compare AROS with LEACH and its variants because they also send data periodically to the base station.

7.3 AROS

AROS is based on clusters with a base station (BS) with “unlimited” energy and “enough” bandwidth in the backbone channels, see Figure 7.1. The BSs are connected to each other by wire, wirelessly or both. To be able to turn off the radio of the sensor nodes as long as possible, we propose to use TDMA to schedule the communication of the sensor nodes. Furthermore, we propose to build clusters where the BSs are the masters in the network. Further, when using clusters we can aggregate data to minimize the communication in the network. The BS can reach all its sensor nodes directly and a similar TDMA scheme as used in LEACH could be used in AROS.

All clusters have a Cluster Head (CH) that can aggregate and fuse data received from sensor nodes in its cluster. CHs are the only sensor nodes that send and forward data to the BS. All CHs may not be able to communicate directly with the BS. Some CHs need other CHs in order to forward the traffic to the BS. For example, CH B in Figure 7.1 is located on the fringe area, and its radio power does not reach the BS. CH B needs to use CH A to forward its traffic. CH B in its turn has to help CH C with forwarding of traffic. Thus, we propose an asymmetric topology where the BS reaches all its sensor nodes while the sensor nodes might not reach the BS directly.

The BS will make route decisions and manage topology changes for its sensor nodes. The BS will construct a TDMA schedule for its sensor nodes and provide the information to each sensor node about their assigned time slot. The BS will look at other BS schedules and ensure that its sensor nodes do not interfere with adjacent sensor nodes. The sensor nodes only need to focus on their own tasks and thereby save energy that otherwise would be used to, e.g., do extra computations or exchange messages with other sensor nodes, in order to maintain the network topology. The BS will change existing routes to save highly exposed sensor nodes from draining their batteries. When a BS receives a message from a new sensor node, it assigns that node to the most suitable BS. When a BS is assigned a new sensor node, the BS will compute the best route and inform any other concerned sensor nodes about the changes. The BS will also check if the network would benefit from rearranging old routes to new ones. No, or little, knowledge of the network is needed at the sensor nodes. The BS can make optimizations that a pure sensor node network would not consider cost-effective. Issues to be considered by the BS include:

- Mobility: Mobile sensor nodes will make the scheduling decisions more complex.
- Energy: When is it worth to reroute the traffic in order to save energy?
- Optimization: What are the network optimization goals and when do we execute the optimizations?
- New sensor nodes/dead sensor nodes: When to do rerouting and optimizations when a new node enters the cluster or demises?
- New sensor nodes added to the network: Which BS does the sensor node try to send its join request to? Does a sensor node need help from other sensor nodes with forwarding of its whereabouts to the BS?
- Timing issues: After what time can a new sensor node be guaranteed to be inserted into a cluster?
- What happens if a BS disappears or a new BS enters the network?

Depending on the TDMA scheme used, the maximum allowed clock skew will be known. From this, and from knowledge about the drift of the local clocks, the maximum time interval between clock synchronizations can be calculated. This in turn implies a maximum sleep time for the sensor nodes, i.e. how often they must listen to the radio in order to keep their clocks in synchronization with the TDMA schedule.

Some sensor nodes in the network could be scheduled for optimized energy saving, while others could be scheduled for Quality of Service (QoS). In our architecture, we can handle sensor nodes with different demands without e.g., involving the whole sensor network for reorganization. The BS will handle all the extra workload, and only the sensor nodes concerned will have to be rescheduled or reclustered. Depending on the application running on the sensor node, i.e. the requested QoS, the BS will schedule the sensor nodes differently. A sensor node with low QoS demands could/would be scheduled to sleep during several TDMA cycles. Sensor nodes with higher demands could/would be scheduled every TDMA cycle (or more often if necessary). Having sensor nodes with low QoS sleep during several TDMA cycles will increase the delay for topology changes and messages from the sensor nodes to the BS. Different QoS demands in the network imply high complexity. Sensor nodes within a cluster must be grouped in a smart way to e.g., guarantee response time.

7.4 Simulations

In order to verify our assumptions that forwarding will reduce the amount of energy for large network sizes, we have set up a fixed, single BS, network in NS 2 [11], created with the centralized cluster formation algorithm that LEACH-C uses, see Section 7.2. The BS does not make any optimizations such as i.e., recalculation of the best cluster formation, or the optimal sleep time. Below we show that AROS, with asymmetric communications and forwarding of packets, extends the lifetime of LEACH and its variants with respect to the amount of energy consumed by the sensor node per data packet sent to the BS. Here we assume that the sensor nodes are clock synchronized, and the BS know the position of the sensor nodes.

We have set up the system using the MIT uAMPS LEACH ns Extensions (uAMPS) [12]. uAMPS was developed on the Network Simulator platform (NS 2) [11]. Test simulations were performed to verify the implementation of LEACH and LEACH-C protocols. We have implemented the LEACH-F protocol in NS 2 and the results were verified based on the simulation results in [2].

First, the simulations were configured as in [2] i.e., a network size of

Table 7.1: Characteristics of the network

	1:st simulation	2:nd simulation
Network size	$100X100$ m	$400X400$ m
BS location, x,y	$50, 175$	$200, 475$
Nodes	100	100
Processing delay	50 μs	50 μs
Radio speed	1 Mbps	1 Mbps
Data size	500 bytes	500 bytes

100x100 meters with 100 nodes randomly distributed and the base station located at position $x = 50$, $y = 175$. That is, the BS was placed 75 meters outside the area where the sensor nodes were deployed. The BS reschedules the CHs every 20:th second. Each node sends a message to its CH during a given time slot. According to [2], the most energy efficient cluster formation have between 3 to 5 clusters in a 100x100 meter network. In order to be able to study the behavior of forwarding, we have chosen to use 4 clusters. We placed 2 clusters close to the BS, to forward data from the 2 clusters placed at the back of the network. The sensor node starts with 2 Joules of energy and the simulation continues until all the sensor nodes in the network have consumed all of their energy. All sensor nodes have an equal amount of energy when the simulation starts. In order to make comparisons possible, we have used the same channel propagation model, radio energy model and beam forming energy model as in LEACH [2]. The energy consumption of the radio transmitter is according to [2] $\epsilon_{friss-amp} = 10pJ/bit/m^2$ for distances under 87 meters and $\epsilon_{two-ray-amp} = 0.0013pJ/bit/m^4$ for distances over 87 meters. The radio electronics cost/energy was set to $E_{elec} = 50nJ/bit$. The data size was 500 bytes/message plus a header of 25 bytes, $b = (500bytes + 25bytes) * 8 = 4200bits$. The equation for calculating the amount of energy used for sending a message d meters is:

$$E_{Tx} = \begin{cases} b * E_{elec} + b * \epsilon_{friss-amp} * d^2 & : d < 87m \\ b * E_{elec} + b * \epsilon_{two-ray-amp} * d^4 & : d \geq 87m \end{cases} \tag{7.1}$$

and the amount of energy used when receiving a message is:

$$E_{Rx} = b * E_{elec} \tag{7.2}$$

Further, all the parameters, such as radio speed, processing delay and radio propagation speed were the same as in [2], see Table 7.1. The energy model can benefit from improvements, however this is outside the scope of this paper.

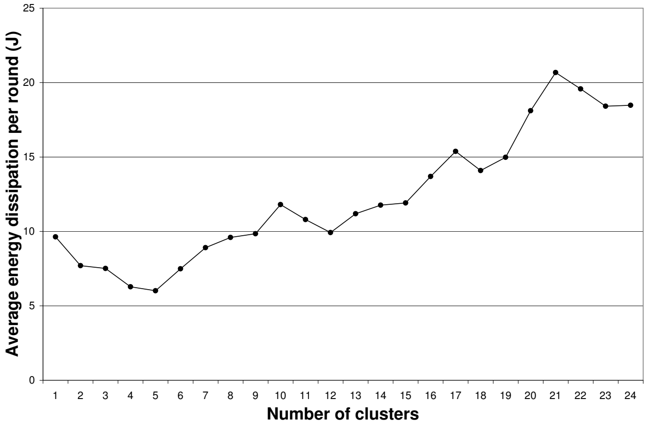

Figure 7.2: Simulation results showing the average energy dissipation per TDMA-round in LEACH.

In the second simulation, the network size was increased to 400x400 meters. The amount of sensor nodes randomly distributed in the network remained the same as in the first simulation, i.e. 100 nodes. Also in this case, we placed the base station 75 meters outside the monitored area, at location $x = 200$, $y = 475$. According to the equation in [2], the optimal number of clusters for this network size is somewhere between 1 and 24 clusters, considering the energy consumption. Simulations with LEACH show that the most energy-efficient cluster formation is between 4 and 5 clusters, see Figure 7.2. In order to study the behavior of forwarding, we have chosen to use an even number of clusters. We put half of the clusters in the front and the other half in the back of the network, from the BS' point of view. The clusters in the back of the network use the clusters in the front to forward their data to the BS. When using even number of clusters, the lowest amount of energy is consumed when using 4 clusters, as can be seen in Figure 7.2. All the parameters, except the BS' location and the network size, are the same as in the first simulation setup, see Table 7.1.

We used LEACH-C's centralized cluster formation algorithm to create the clusters in AROS. The clusters were then manually changed to better suit 4 clusters with forwarding. It is not always the case that the clusters generated by the centralized cluster formation algorithm create cluster formations where forwarding of data can be studied. In some cases it creates one cluster far

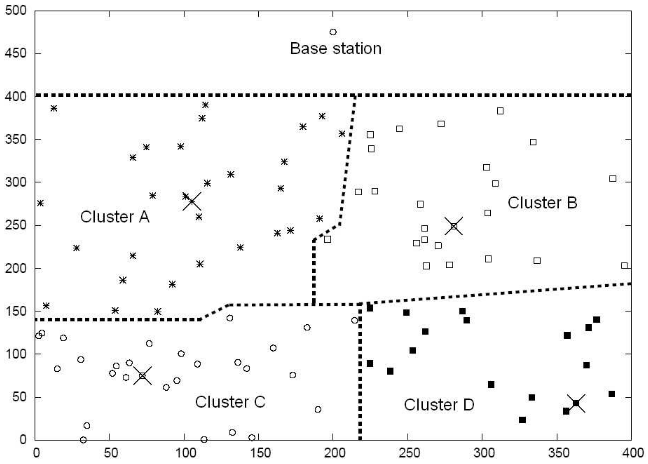

Figure 7.3: Cluster formation of the simulated network using 4 clusters and a network size of 400x400 meters.

away from the BS and three clusters beside each other nearby the BS. This was the case when trying to create a suitable cluster formation for AROS using 4 clusters. However, earlier simulations in LEACH-C with 5 clusters showed a cluster formation suitable for 4 clusters when 3 of the clusters where merged into 2. This cluster formation is also used for LEACH-F in order to simulate the same cluster scenario.

The sensor nodes are scheduled to send their data to a cluster head during a given slot. The cluster heads furthest away from the BS i.e., Cluster C and Cluster D, see Figure 7.3, were modified to send their aggregated data to the cluster heads in Cluster A and Cluster B respectively, instead of sending it directly to the BS. The cluster heads in Cluster A and Cluster B forwards the aggregated data directly to the BS after receiving it.

The length of the TDMA cycle for a cluster depends on how many nodes there are in the cluster. The length of the TDMA cycle is updated every 20:th second, at the same time as the network is rescheduled. Cluster A and Cluster C might have different TDMA cycle length, due to different number of nodes in the cluster. To simplify the forwarding schedule, we used the longest TDMA cycle of Cluster A and Cluster C, plus some overhead, as cycle lengths for Cluster A and Cluster C. The same cycle lengthening was done between Cluster B and Cluster D.

7.5 Results

The results from our experiments with a 100x100 meter scenario, show that AROS perform almost as well as LEACH-C and LEACH-F, depicted in Figure 7.4. In spite of the fact that the CHs in AROS send the data a shorter way towards the BS, the extra receive and send when forwarding data sometimes use more energy than to send it directly to the BS. AROS sends almost as much data to the BS as LEACH-C and LEACH-F. The data from the clusters furthest away has a longer delay time before the BS receives the data. This is due to the prolonged TDMA-cycle of the smaller cluster, see Section 7.4, and due to the extra hop the data needs to travel. AROS will perform even better when optimizing the cluster formations, data routing and the TDMA-schedule.

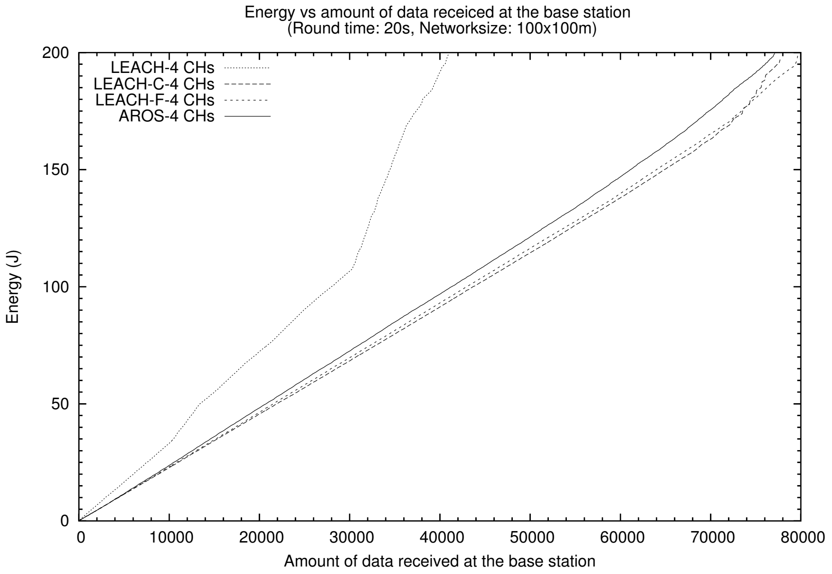

Figure 7.4: Total data received at the base station per given amount of energy in a 100x100 m large network with 4 clusters.

When the network was increased to 400x400 meters, LEACH did not perform well. The nodes furthest away from the BS demised early and data from that area could not be received at the BS. The early drop out of the nodes were due to the radio transmission, draining the node when they were trying to send data to the BS. AROS, on the other hand, handles this by sending its data shorter distances. The total amount of energy consumed, E_{tot}, when sending a message to the BS depends on the number, n, of forwarding CHs between the sending CH and the BS. Equation (7.1) and (7.2) are used to calculate the total energy consumed E_{tot} as:

$$E_{tot} = \begin{cases} E_{Tx_n} & : n = 0 \\ E_{Tx_0} + \sum_{k=1}^{n}(E_{Rx_k} + E_{Tx_k}) & : n > 0 \end{cases} \tag{7.3}$$

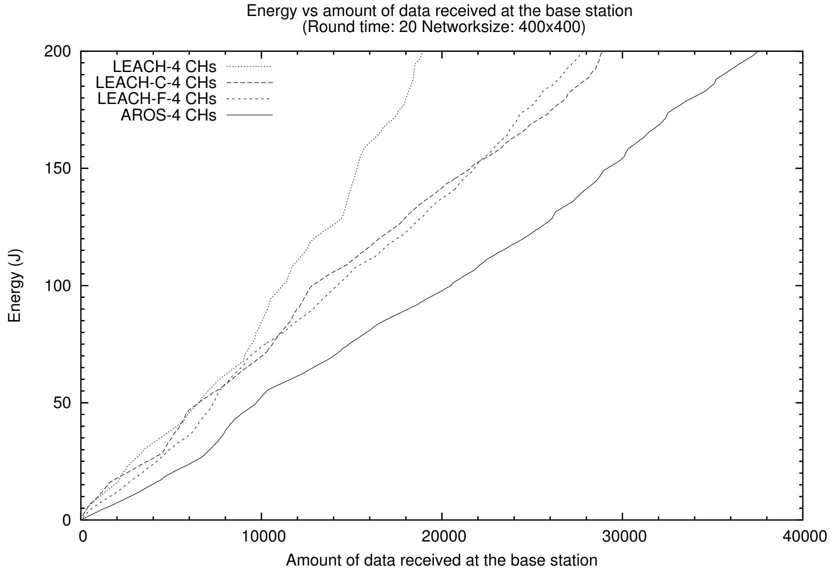

Figure 7.5: Total data received at the base station per given amount of energy in a 400x400 m large network with 4 clusters.

For example, consider a sending CH located 475 meters from the BS. The amount of energy consumed in LEACH, to send data to the BS is $E_{tot_{LEACH}} \approx 278mJ, n = 0$ (7.3). The amount of energy consumed when using AROS with one forwarding CH is $E_{tot_{AROS}} \approx 53mJ$ (7.3). The CH that forwards the data in this example is located half-way between the BS and the sending CH, $d = 237,5m$. As one can see, LEACH consumes more than five times more energy than AROS.

Table 7.2: Data received at base station per unit energy (J)

Protocol	Data Packets/Energy (J)	AROS is
LEACH	$\frac{19160}{204.2} \approx 93.8$	97% better
LEACH-C	$\frac{29240}{202.2} \approx 144.6$	28% better
LEACH-F	$\frac{28581}{203.7} \approx 140.3$	32% better
AROS	$\frac{37979}{205.2} \approx 185.1$	

When comparing how much data the BS receives per Joule of energy in Table 7.2, we can see that AROS performs 97% better than LEACH, 28% better than LEACH-C and 32% better than LEACH-F. This is also depicted in Figure 7.5.

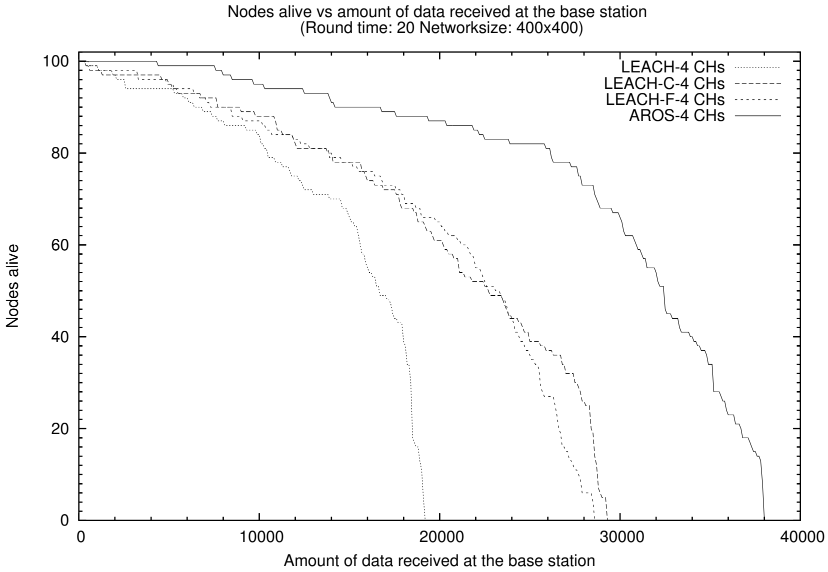

Figure 7.6: Number of nodes alive compared to the amount of data received at the base station in a 400x400 m large network with 4 clusters.

Figure 7.5 also shows that when LEACH-C and LEACH-F have used all of its energy and demises, AROS still has 25% of its energy left and 54% of its energy left when LEACH demises. In Figure 7.6 we can see that AROS has more than 73% of its nodes alive when LEACH-F has zero nodes alive in the network. When LEACH-Cs network demises AROS has 68% of its nodes alive and if we compare to LEACH, AROS has approximately 88% of its nodes alive. This results in a situation where the BS can receive at least 9000 more messages from the network before all energy is consumed.

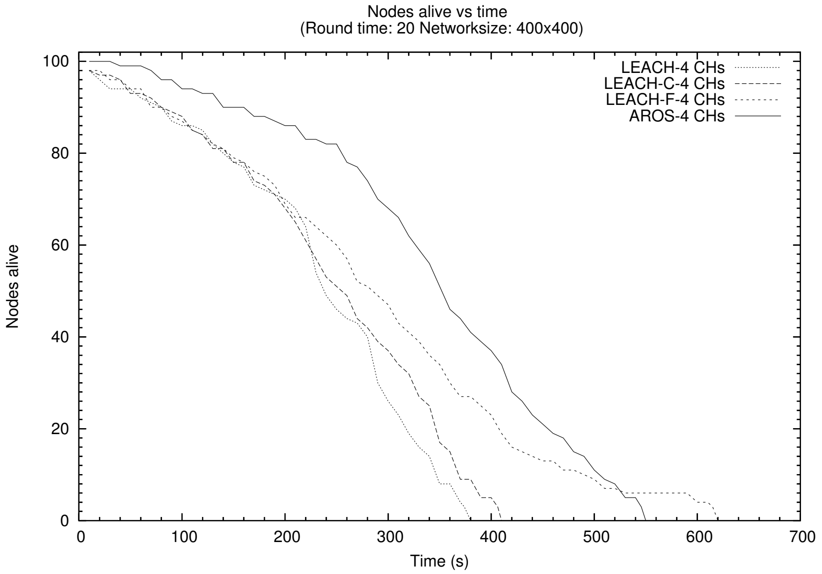

Figure 7.7: The amount of nodes alive over time in a 400x400 m large network with 4 clusters.

The energy consumed in the network is evenly distributed among the nodes in AROS. Clusters far away from the BS in AROS will survive until the end and continue to gather information. In contrast to LEACH-F where only the clusters closest to the BS are alive at the end and the clusters far away are demised, see Figure 7.7. At time 340, when Cluster D in LEACH-F is demised, LEACH-F has only 40% of its nodes left in the network. AROS on the other hand still has 61% of its nodes left in Cluster D and 56% of its nodes left in the network. This implies that AROS still can collect data from the whole network area but LEACH-F can not because one cluster has demised. At time 400, when LEACH and LEACH-C demises, AROS still collects data from the whole network with 28% of the nodes left in Cluster D, 30% of the nodes left in Cluster A, 29% of the nodes left in Cluster B and 54% of the nodes left in Cluster C. LEACH-F can only collect data from Cluster A, B and C with 20%, 29% respective 36% of its nodes left alive. Until time 440, AROS is able to collect data from the whole network with nodes alive in all 4 clusters. This is 30% longer time than with LEACH-F that only collects data from 3 clusters, Cluster A, B and C. At time 540 LEACH-F has one cluster left alive, Cluster A, with 6 nodes very close to the BS. AROS has 2 clusters left, Cluster A and C, with 2 respective 3 nodes left alive.

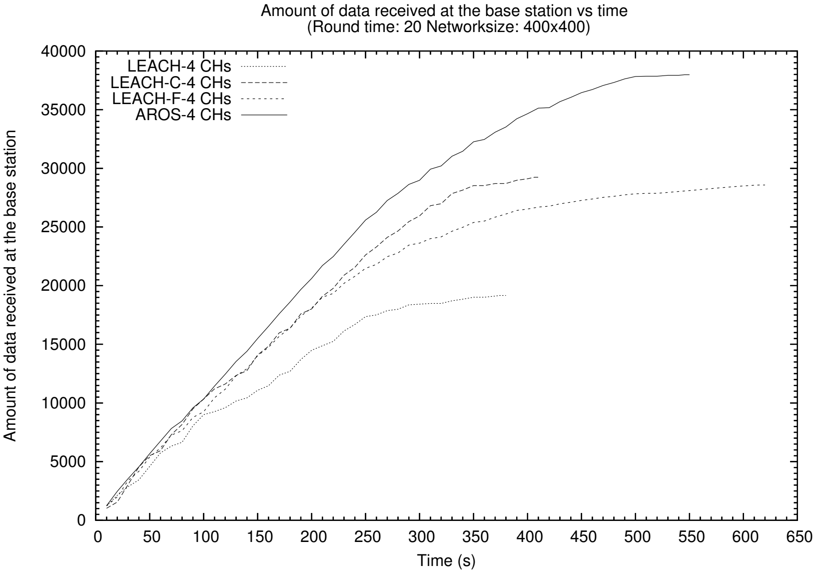

Figure 7.8: Amount of data received at the base station over time in a 400x400 m large network with 4 clusters.

Reducing the energy consumption for sending data, each nodes' lifetime is prolonged and more data can be sent to the BS, as showed in Figure 7.8. This can also be seen in Figure 7.5, the total data received at the BS per given amount of energy. As a result for having more nodes alive AROS can gather more data from a larger network area.

If we compare AROS and LEACH-F at time 340 again, when the first cluster demises in LEACH-F, we can se that AROS gathers 80% more data until the whole network demises. When looking at the time after AROS has demised, LEACH-F only gathers 468 messages during the last 75 seconds, and that data is only from one cluster closest to the BS, as mentioned earlier. At time 500 LEACH-F has almost no energy left and the few nodes left in the last cluster sends very few messages, see Figure 7.9. This means that LEACH-F prolongs the network lifetime collecting data from a very small area. Even though LEACH-F lives slightly longer, AROS collects data from sensors widely spread over a larger network area during its whole lifetime.

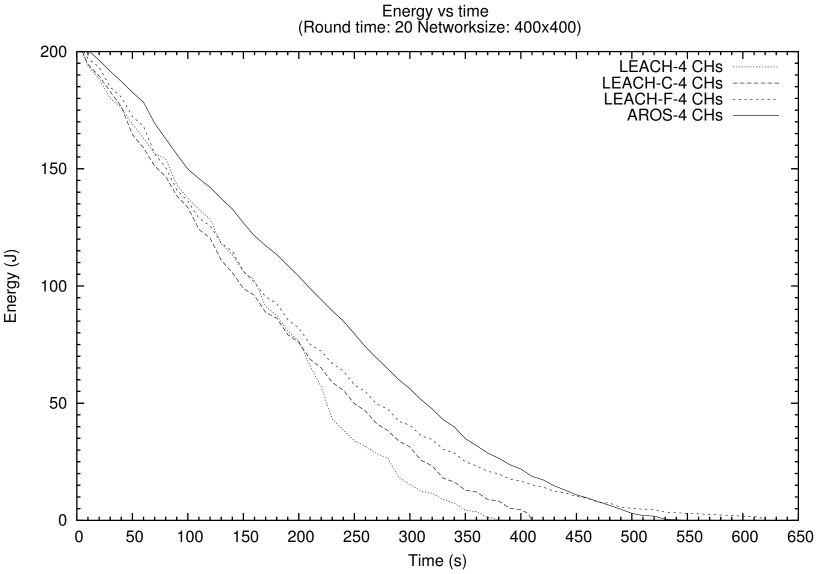

Figure 7.9: Energy left in the system over time in a 400x400 m large network with 4 clusters.

7.6 Conclusions

We have presented a simulation comparison between asymmetric and symmetric communication in sensor networks. In the simulation studies, we have compared AROS, which uses asymmetric communication, to LEACH and its two variants, LEACH-C and LEACH-F.

In AROS, a base station acts as a master for the sensor nodes and can reach all its sensor nodes in one hop. However, all sensor nodes might not reach the base station in one hop. In order to minimize the communication between the sensor nodes, the base station will do route decisions and manage topology changes. The base station will also make a TDMA schedule for its sensor nodes and inform each sensor node about their assigned time slot. In this paper, the base station does not make any optimizations such as e.g., recalculation of the best cluster formation, or sleep time. AROS is similar to LEACH, a cluster based protocol where the clusters have cluster heads that can aggregate and fuse data received from the sensor nodes in its cluster.

All sensor nodes start with a fixed amount of energy and the simulation continues until all the sensor nodes in the network have consumed all of their energy. The simulations have shown that AROS extends the lifetime of the LEACH protocols in large networks and that AROS performs almost as well as the LEACH protocols in small networks.

In these simulations we have not used any advanced features of the base station (such as e.g., reclustering and rescheduling). Instead we have studied static network configurations. Still, we have shown that AROS is significantly better than LEACH and its variants in collecting data to a base station with the same total amount of energy. Because the energy consumed in the AROS network is evenly distributed among the nodes, AROS can collect data from sensors widely spread over a larger network area. Clusters far away from the BS will live longer and continue to gather information until the end. AROS has 25% of its energy left when the other LEACH protocols have used all of their energy and demised. We have shown, after sending the same amount of data to the BS, that AROS has more than 73% of its nodes alive when LEACH-F has zero nodes alive in the network.

The simulations presented in this paper were performed in order to show that asymmetric communication with multihop extends the lifetime of the sensor nodes in large networks. Optimizations and more complex TDMA scheduling will be investigated in future work.

Our next step is to design a TDMA scheduler for AROS multihop networks and a base station implementation in NS in order to make dynamic simulations. The TDMA scheduler will optimize the network for energy saving, cluster formations and routing. Further, we will evaluate what types of scenarios AROS is suitable for.

Bibliography

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Maui, Hawaii, Jan 2000. In Proceedings of the 33rd International Conference on System Sciences (HICSS '00).

[2] W. Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts institute of technology, June 2000.

[3] J. Neander, E. Hansen, M. Nolin, and M. Björkman. An asymmetric network architecture for sensor networks. Technical Report ISSN 1404-3041 ISRN MDH-MRTC-181/2005-1-SE, Department of Computer Science and Electronics, Malardalens University, August 2005.

[4] J. Neander, M. Nolin, and M. Björkman. Using Existing Infrastructure as Proxy Support for Sensor Networks. *16th EUROMICRO Conference on Real-Time Systems (ECRTS 04), WiP*, Jun 2004.

[5] M. Gerla, T.J. Kwon, and G. Pei. On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering. *Procedings of IEEE WCNC 2000*, Sep 2000.

[6] G. Pei and C. Chien. Low Power TDMA in Large Wireless Sensor Networks. *Military Communications Conference*, 1:347–351, 2001.

[7] G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[8] T. Maruta and H. Ishibuchi. Performance Evaluation of Genetic Algorithms for Flowshop Scheduling Problems. *Proceedings of the 1st IEEE Conference on Evolutionary computation*, 2:812–817, June 1994.

[9] A. Manjeshwar and D. P. Agrawal. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. *Parallel and Distributed Processing Symposium., Proceedings 15th International*, pages 2009–2015, April 2001.

[10] A. Manjeshwar and D. P. Agrawal. APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks. *Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002*, pages 195–202, April 2002.

[11] NS 2, Network Simulator. World Wide Web, `http://www.isi.edu/nsnam/ns/ns-build.html`, 2004.

[12] MIT uAMPS LEACH ns Extensions. World Wide Web, `http://www.ece.rochester.edu/research/wcng/code/index.htm`, 2005.

Chapter 8

Paper B: Energy-Efficient Cluster Formation for Large Sensor Networks using a Minimum Separation Distance

Ewa Hansen, Jonas Neander, Mikael Nolin and Mats Björkman

In proceedings of the Fifth Annual Mediterranean Ad Hoc Networking Workshop 2006, MedHocNet, Lipari, Italy, June 2006

Abstract

In this paper we investigate the usefulness of enforcing a minimum separation distance between cluster heads in a cluster based sensor network, thereby prolonging network lifetime by spreading the cluster heads, thus lowering the average communication energy consumption.

We have performed initial simulations in order to determine how much we can lower the energy consumption in the sensor network by separating the cluster heads. We have also investigated how the number of clusters affect the energy consumption for a given minimum separation distance.

The results show that our sensor network performs up to 150% better when introducing a minimum separation distance between cluster heads, comparing the number of messages received at the base station. The simulations also show that the minimum separation distance resulting in the lowest energy consumption in our network varies with the number of clusters.

8.1 Introduction

The need for energy-efficient infrastructures for sensor networks is becoming increasingly important. Wireless sensor networks are networks consisting of many sensor nodes that communicate over a wireless medium. A sensor node is equipped with a sensor module, a processor, a radio module and a battery. Since the battery limits the lifetime of the sensor nodes it also limits the lifetime of the sensor network, thus energy efficiency is a major issue for sensor networks.

An important goal in many sensor networks is to monitor an area as long time as possible. Hence, it is important to distribute energy consumption evenly across the network. When energy consumption is evenly distributed, the major part of the sensor nodes will stay alive approximately the same amount of time. This enables continued information gathering throughout the whole network area during the lifetime of the network.

The most power-consuming activity of a sensor node is typically radio communication [1], this applies to transmission and reception, and also to listening for data. Hence, radio communication must be kept to an absolute minimum. This means that the amount of network traffic should be minimized. In order to reduce the amount of traffic in a network, we can build clusters of sensor nodes as proposed in e.g. [2, 3, 4]. Some sensor nodes become cluster heads and gather all traffic from their respective cluster. The cluster head aggregates or fuses the gathered data and then sends it towards the base station. When using clustering, the workload on a cluster head is larger than for non-cluster heads. The cluster heads should therefore be changed several times during the lifetime of a sensor network in order to distribute the extra workload and energy consumption evenly.

Our hypothesis is that the geographical distribution of the cluster heads severely influences the overall energy consumption of the network. Spreading the cluster heads more evenly means prolonging the lifetime of the network. Simulation results presented in this paper indicate that introducing a minimum separation distance between cluster heads improves network lifetime.

For our simulations we have used the AROS architecture, Asymmetric communication and ROuting in Sensor networks [5]. AROS is an extension of LEACH-C [6] which is a well known cluster-based sensor network architecture. The AROS architecture is based on cluster groups using base stations with “unlimited” energy and “enough” bandwidth in the backbone network. In AROS we use a centralized approach where the resource-adequate base stations perform all the calculations necessary to evaluate routes and schedules, thus relieving sensor nodes from the energy-consuming task of executing complex distributed decision algorithms. Often, the base stations can be situated in existing infrastructures. For instance, there are infrastructure networks built in hospitals and industrial factories that could be used to host base stations. The infrastructure network can act as a, possibly fault tolerant, base station backbone for sensor nodes gathering data or monitoring patients.

In order to be able to turn off the radio of the sensor nodes as long as possible to save energy, we use Time Division Multiple Access (TDMA) to schedule the communication of the sensor nodes. Furthermore, we use clusters to ease the scheduling of the sensor nodes. When using clusters we can aggregate or fuse data to lower the communication needs in the sensor network.

AROS is based on clusters where the cluster heads gather data from their cluster nodes and then transmit it to the base station. AROS has an asymmetric topology where the base station is able to transmit information to all its sensor nodes directly. All cluster heads may however not be able to transmit data directly to the base station. Hence, traffic from these cluster heads must be routed through other cluster heads in order to reach the base station. However, routing of traffic through other cluster heads will increase the power consumption of the forwarding cluster heads. Therefore, routing decisions must be carefully evaluated in order to maximize network lifetime.

In our simulations we have experimented with a minimum separation distance between cluster heads. We have also investigated how the number of clusters used, together with this minimum separation distance, affects the energy consumption in the network. The minimum separation distance is the smallest distance allowed between cluster heads. The distance can be larger than the minimum separation distance but should not be smaller. The simulations were performed in order to investigate the effects on energy consumption when using a minimum separation distance between cluster heads.

The simulations show that the minimum separation distance resulting in the lowest energy consumption in our network varies with the number of clusters. The simulations also show that it is up to 150% better to use a minimum separation distance between cluster heads than not using any minimum separation distance at all, measured by the number of messages received at the base station. By using a minimum separation distance between cluster heads we can make the network gather more messages from the network for a longer period of time.

The rest of this paper is outlined as follows: in Section 8.2, we describe some related work. In Section 8.3, we present the minimum separation distance algorithm and the simulation setup. In Section 8.4 we present the results from our simulations, and finally, in Section 8.5 we present our conclusions.

8.2 Related Work

LEACH (Low-Energy Adaptive Clustering Hierarchy) [3] is a TDMA cluster based approach where a node elects itself to become cluster head by some probability and broadcasts an advertisement message to all the other nodes in the network. A non cluster head node selects a cluster head to join based on the received signal strength. Being cluster head is more energy consuming than being a non cluster head node, since the cluster head needs to receive data from all cluster members in its cluster and then send the data to the base station. All nodes in the network have the potential to be cluster head during some periods of time. The TDMA scheme starts every round with a set-up phase to organize the clusters. After the set-up phase, the system is in a steady-state phase for a certain amount of time. The steady-state phase consists of several cycles where all nodes have their transmission slots periodically. The nodes send their data to the cluster head that aggregates the data and sends it to its base station at the end of each cycle. After a certain amount of time, the TDMA round ends and the network re-enters the set-up phase.

LEACH-C (LEACH-Centralized) [6] is a variant of LEACH that uses a centralized cluster formation algorithm to form clusters. The protocol uses the same steady-state protocol as LEACH. During the set-up phase, the base station receives information from each node about their current location and energy level. After that, the base station runs the centralized cluster formation algorithm to determine cluster heads and clusters for that round. LEACH-C uses simulated annealing [7] to search for near-optimal clusters. LEACH-C chooses cluster heads randomly but the base station makes sure that only nodes with “enough” energy are participating in the cluster head selection. Once the clusters are created, the base station broadcasts the information to all the nodes in the network. Each of the nodes, except the cluster head, determines its local TDMA slot, used for data transmission, before it goes to sleep until it is time to transmit data to its cluster head, i.e., until the arrival of the next slot.

A further development is LEACH-F (LEACH with Fixed clusters) [6]. LEACH-F is based on clusters that are formed once - and then fixed. Then, the cluster head position rotates among the nodes within the cluster. The advantage with this is that, once the clusters are formed, there is no set-up overhead at the beginning of each round. To decide clusters, LEACH-F uses the same centralized cluster formation algorithm as LEACH-C. The fixed clusters in LEACH-F do not allow new nodes to be added to the system and do not adjust their behavior based on nodes dying.

BCDCP (Base-station Controlled Dynamic Clustering Protocol) [8] is a centralized routing protocol with a high-energy base station that makes all the high energy-consuming activities e.g. selecting cluster heads and routing paths, performing randomized rotation of cluster heads. The idea in BCDCP is to organize balanced clusters with uniform placement of cluster heads where each cluster head serves an approximately equal number of member nodes.

During each setup phase the base station receives information on the current energy status from all the nodes in the network. BCDCP uses an iterative splitting algorithm to form clusters. The first step is to choose two nodes, among the eligible nodes, that have the maximum separation distance. Step two is to group the remaining nodes to one of the cluster heads, whichever is closest. Step tree is to balance the clusters so that each cluster has approximately the same number of nodes. Step four is to start from step one and split the sub-clusters in to smaller parts. The iteration of the four steps continues until the desired number of cluster heads is attained.

8.3 Our Approach

In order to be able to see the effects on energy consumption when using a minimum separation distance between cluster heads, we have developed a simple algorithm to find and select cluster heads.

8.3.1 Cluster head selection algorithm

In our cluster formation algorithm, we use the same simulated annealing as LEACH-C to minimize the energy consumption for cluster nodes when transmitting data to the cluster head. As LEACH-C, we randomly choose a node among the eligible nodes to become cluster head but we also make sure that the nodes are separated with at least the minimum separation distance (if possible) from the other cluster head nodes.

In the cluster head selection part, see Figure 8.1, cluster heads are randomly chosen from a list of eligible nodes. To determine which nodes are eligible, the average energy of the remaining nodes in the network is calculated. In order to spread the load evenly, only nodes with energy levels above average are eligible.

```
MSD = Minimum Separation Distance
dc = Number of desired cluster heads,
energy(n) = Remaining energy for node n
avg = $\frac{\sum energy(n)}{number\ of\ alive\ nodes}$
eligible $= \{$n$|$ energy(n) $\geq$ avg $\}$
assert($|$eligible$| \geq$ dc)
CH= { }

While ($|$CH$| <$ dc)
    if $\exists$n: n$\in$eligible $\bigwedge$ ($\forall$ m$\in$CH, dist(m,n)) $\geq$ MSD
        add(n , CH)
        remove(n , eligible)
    else
        n $\in$ eligible
        add(n, CH)
        remove(n, eligible)
```

Figure 8.1: **Algorithm to select Cluster Heads (CH)**

If a node that has been randomly chosen is too close i.e. *within* the range of the minimum separation distance from all other chosen cluster heads, a new node has to be chosen to guarantee the minimum separation distance. This process iterates until the desired number of cluster heads is attained. If we cannot find a node outside the range of the minimum separation distance (to guarantee the minimum separation distance) we choose any node among the eligible nodes to become cluster head[1].

When all cluster heads have been chosen and separated, generally with at least the minimum separation distance, clusters are created the same way as in [6].

8.3.2 Simulation Setup

In the performed simulations we have varied the minimum separation distances between cluster heads, in order to see the effects on energy consumption in the network. We have also investigated whether the number of clusters used, together with the minimum separation distance, has any effect on the energy

---

[1] The algorithm is simplified for these simulations, i.e. the assert in Figure 8.1 will always be true.

consumption. The minimum separation distance varied between 50 and 140 meters, and the number of clusters varied between 2 and 15 clusters.

All simulations presented in this paper were performed within one network setup. That is, we have used the same number of nodes and the same position of these nodes in all experiments presented in the paper.

The simulations where performed in the network simulator NS 2 [9], using a network size of 400x400 meters where 100 sensor nodes were randomly distributed in the network. In the simulations we assume that the sensor nodes are static. We placed the base station 75 meters outside the monitored area, at location $x = 200$, $y = 475$. All sensor nodes start with a fixed amount of energy and the simulation continues until all the sensor nodes in the network have consumed all of their energy. Since AROS is an extension of LEACH, we have used the same simulation setup and radio model for our simulations as in LEACH [6], and all other parameters such as radio speed, processing delay and radio propagation speed were the same as in [6, 5].

In [6], Heinzelman has calculated how often the cluster heads should be changed, i.e. the round-time. The calculation was made for a 100x100 meters network. Due to the larger energy consumption of sending longer distances in a 400x400 meters network, we need to change cluster heads more often than every 20:th second, which is the round-time for the 100x100 meters network [6]. In our simulations we change cluster heads every 10:th second. This is a tradeoff between rescheduling cost, efficiency and energy consumption balance. When the network reschedules new cluster heads are chosen and new clusters are formed.

8.4 Results

In Figure 8.2, we see how the minimum separation distance affects the energy consumption, i.e., the number of messages received at the base station during the lifetime of the network. We also see how the number of clusters used affects the energy consumption in the network. In the same figure we see that when using 2 clusters, the number of messages received at the base station is low in all our simulations. Further, we see that when using 4 clusters and a minimum separation distance of 130 meters between cluster heads, the base station receives the most messages. It is not always the case that 4 cluster yield the most messages to the base station. For some minimum separation distances 3 cluster heads yields the most messages. Below, we have therefore looked at the simulation results in more detail when using 3 and 4 clusters, respectively.

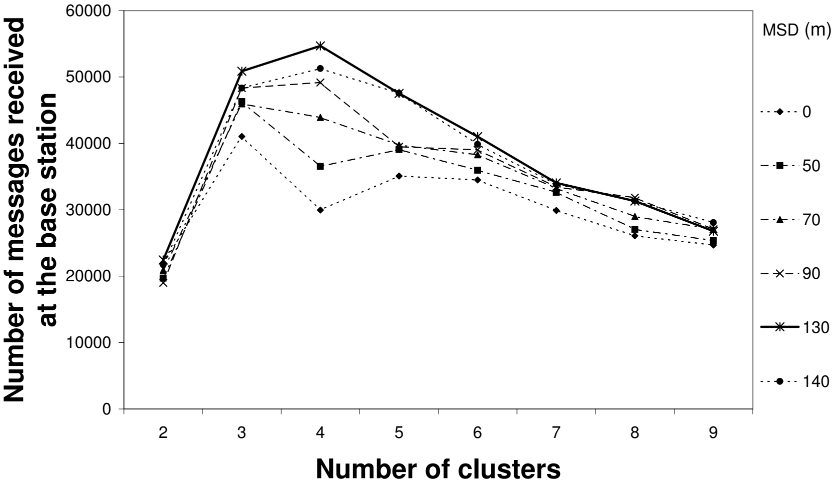

Figure 8.2: Messages received

In Figure 8.2, we see that using a minimum separation distance between cluster heads is better than not to use any to control the placement of the cluster heads. By using a minimum separation distance between cluster heads we can make the network gather more messages from the network for a longer period of time. The figure also shows that a minimum separation distance of 130 meters delivers the most messages to the base station for almost all number of clusters.

8.4.1 Using 3 Clusters

In Figure 8.3, we present simulation results when using 3 clusters. In order to be able to see the curves more distinctively in the figure we have chosen to only show a subset of the curves[2]. In Figure 8.3, we see that when not using a minimum separation distance between cluster heads, the base station receives approximately 41000 messages. However, when using 3 clusters and 130 meters as the minimum separation distance, the base station receives approximately 51000 messages, which is an enhancement of 24%, or 10000 messages.

---

[2] All curves not represented in the figure are located in between the curves MSD: 0 meters and MSD: 130 meters.

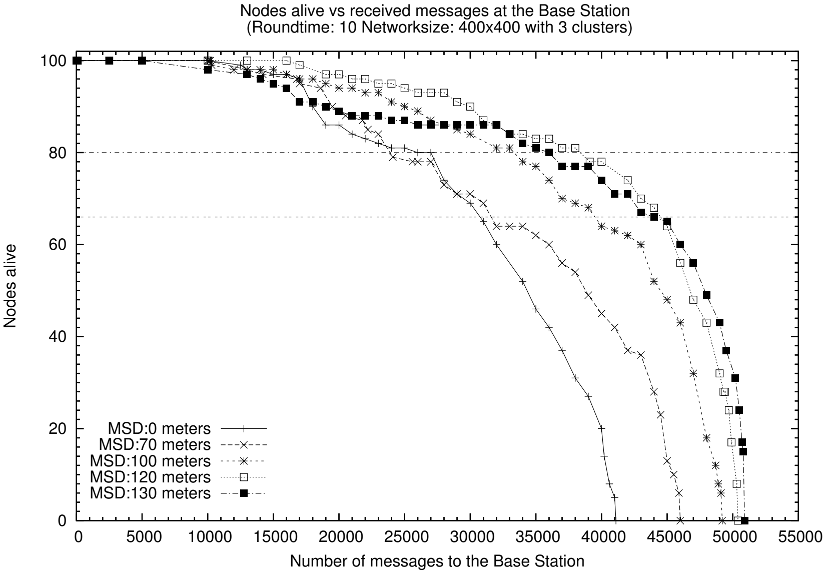

Figure 8.3: Using 3 clusters

If we look at 80% tolerance limit[3], illustrated with the upper horizontal line in Figure 8.3, we see that when not using a minimum separation distance the curve drops below the tolerance limit already at 26000 messages. When using 130 meters as the minimum separation distance the curve drops below the tolerance limit at 37000 messages, while when using 120 meters as the minimum separation distance the curve drops below the tolerance limit at 39000 messages.

Depending of the tolerance limit, different minimum separation distances yield the longest network lifetime, e.g., the crossover point between using 120 and 130 meters as the minimum separation distances is slightly above 65% sensor nodes alive, meaning that for tolerance limits above 65%, using a 120 meters minimum separation distance yields the longest network lifetime (in terms of messages received at the base station). The 65% tolerance limit is illustrated with the lower horizontal line in Figure 8.3.

---

[3] Most sensor networks have a lower limit on the number of nodes that must be alive in order for the network to still be functional, we call this limit the *tolerance limit*.

In general, the spread between the minimum separation distance curves is small in the figure and they all have a rather gradual slope (see also discussion on slope below).

8.4.2 Using 4 Clusters

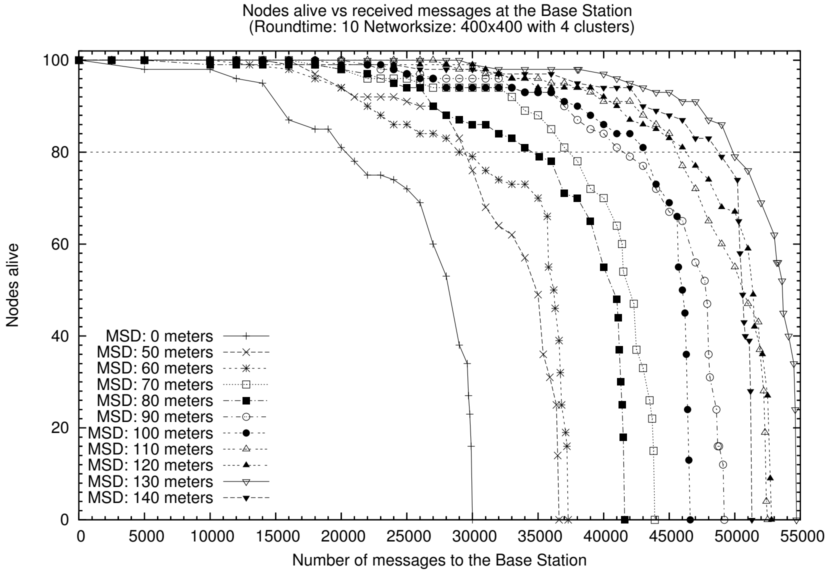

Figure 8.4: Using 4 clusters

In Figure 8.4, we present simulation results when using 4 clusters. We show that when using 4 clusters and a minimum separation distance of 130 meters between cluster heads, the base station receives almost 55000 messages, compared to the simulation with 4 clusters and no minimum separation distance where the base station only receives approximately 30000 messages. The minimum separation distance of 130 meters between cluster heads thus gives an enhancement of 80%, or 25000 messages.

If we look at the 80% tolerance limit, we see that the 130 meters minimum separation distance curve crosses the limit at about 50000 messages, while the 0 meters minimum separation distance curve crosses the limit already at about

20000 messages. Using 130 meters as the minimum separation distance thus gives an enhancement of 150%, or 30000 messages, compared to when not using any minimum separation distance.

When comparing the results from using 3 clusters and 4 clusters, we see that the number of messages received is larger for 4 clusters than for 3 clusters for the best minimum separation distances. We can also see in the figures that the spread between different minimum separation distances is much larger for 4 clusters than for 3 clusters, meaning that the choice of minimum separation distance becomes much more important. It can also be noted that most curves have a steeper slope when using 4 clusters than when using 3 clusters. This means that using 4 clusters can be more advantageous for high tolerance limits. In our figures, when using 130 meters as the minimum separation distance, the total number of messages received is 51000 and 55000 for 3 and 4 clusters, respectively, a relatively small difference, less than 10%. However, when comparing the same curves at the 80% tolerance limit, the number of messages received is 37000 and 50000, respectively. Here, the relative difference is around 30%. The conclusion from this example is that the slope of the curve matters, this will be further discussed below.

8.4.3 Minimum separation distance or not?

Figure 8.6 show results from simulations with a minimum separation distance of 130 meters and the number of clusters varied between 2 and 9. As mentioned above, when using 4 clusters and a minimum separation distance of 130 meters between cluster heads, the base station receives the most messages. When using the tolerance limit of 80%, the base station receives approximately 50000 messages.

The bad performance when using 2 clusters can clearly be seen in Figure 8.6, approximately 8000 messages are received when the 80% tolerance limit is reached. The reason for this is that when using only 2 clusters, the communication distances between nodes become so long that the radio energy consumption (which is super-linear with communication distance) increases very much. It can also be seen in the figure that the slope when using 3 clusters is very gradual, as was observed earlier.

Figure 8.5 shows results from simulations without minimum separation distance, i.e., 0 meters as the minimum separation distance. We can see that when using the 80% tolerance limit and optimizing for maximum number of messages received at the base station, the best configuration of the sensor network is to use 6 clusters. The base station then receives approximately 33000 mes-

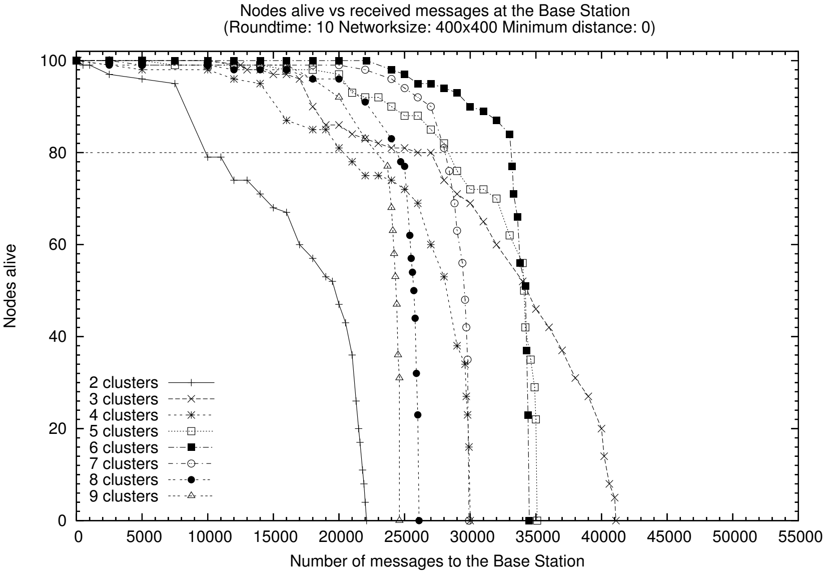

Figure 8.5: No Minimum Separation Distance

sages. When using 6 clusters and a minimum separation distance of 130 meters between cluster heads, depicted in figure 8.6, the base station receive approximately 40000 messages when using the 80% tolerance limit. Using 130 meters instead of not using a minimum separation distance thus yields an enhancement of 7000 messages.

Comparing Figure 8.5 and Figure 8.6 we see that regardless of how many clusters we choose to use in the network, using a minimum separation distance of 130 meters between cluster heads instead of not using any minimum separation distance will make the network stay alive longer and deliver more messages to the base station.

8.4.4 Efficient utilization

Efficient utilization of the energy resources of the sensor nodes will increase the lifetime of the sensor network. In the ideal network, all sensor nodes would live exactly the same period of time.

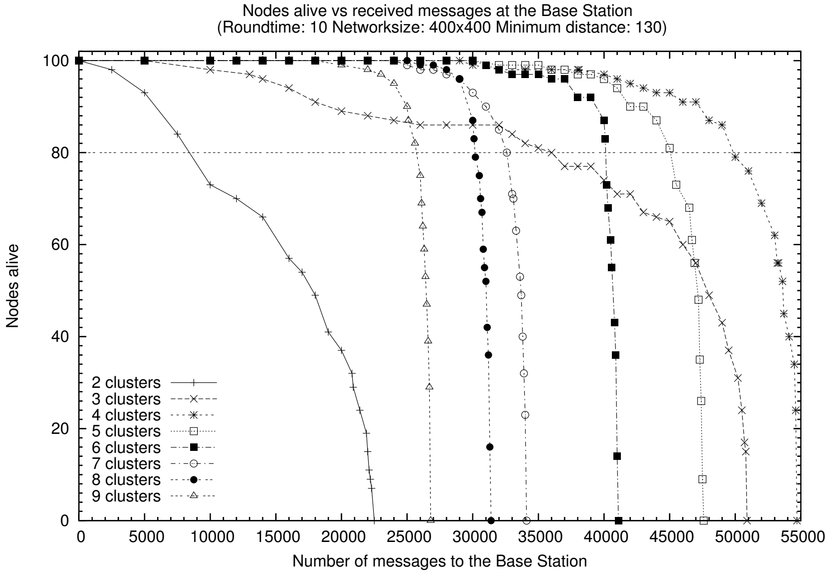

Figure 8.6: Minimum Separation Distance 130 meters

In Figure 8.6, we see that a more efficient utilization of the sensor nodes' power makes the sensor network stay alive a longer period of time. In the figure we can also see that as soon as the sensor nodes in the network start to demise, the whole network demises shortly after, for all number of clusters above 3.

To be able to say that the utilization of the sensor nodes' energy has been efficient, we want the "knee" of the curve to be as sharp as possible, see Figure 8.6. The sharper the knee is, the better the energy consumption is distributed among the sensor nodes.

We want the knee to drop as late as possible and when it finally drops the gradient should be as steep as possible. This indicates that the sensor nodes have been utilized efficiently, hence the network lives longer. This steep gradient also indicates that the whole network area is monitored almost until the whole network demises.

In Figure 8.5, we see a sharp knee and a steep gradient only when using 6 clusters. This indicates that most of the sensor nodes have been utilized efficiently when using 6 clusters. Looking for sharp knees and steep gradients in Figure 8.6, we can see that almost every choice of number of clusters have a steep gradient, except for 2 and 3 clusters, which have a more gradual slope.

When looking at Figure 8.6, we can see that when the number of clusters increases the sharper the knee becomes. Unfortunately this is a tradeoff between sharp knees and the total number of messages received. The figure show that despite of the fact that 8 and 9 clusters have the sharpest knees, using 4 clusters still delivers more messages to the base station at all times. When using 8 or 9 clusters the base station receives totally 31000 and 26000 messages respectively, while when using 4 clusters all nodes are still alive continuing to gather information, when the base station has received the same amount of messages. This means that even though 8 or 9 clusters have the sharpest knees, using 4 clusters is still a better choice, when comparing the number of messages received at the base station.

8.5 Conclusions

In this paper we have presented simulation results from our experiments with a minimum separation distances between cluster heads. We have performed these simulations in order to be able to determine how much we can lower the energy consumption in the sensor network by separating the cluster heads, i.e., by distributing the cluster heads through the whole network.

We have presented a simple energy-efficient cluster formation algorithm for the wireless multihop sensor network AROS.

We have shown that using a minimum separation distance between cluster heads improves energy efficiency, measured by the number of messages received at the base station. We have also shown that it is better, up to 150% in our simulations, to use a minimum separation distance between cluster heads than not to use any minimum separation distance. By using a minimum separation distance between cluster heads we make the network live longer, gathering data from the whole network area. We have also shown that the number of clusters used together with the minimum separation distance affects the energy consumption. Using 4 clusters and a minimum separation distance of 130 meters between cluster heads is the best configuration for our simulated network.

Our simulations have also shown that, depending on the number of dead nodes that can be tolerated, different minimum separation distances as well as different number of clusters affects the number of messages received before the given tolerance limit is reached. Looking at the slope of the curve can give a good feeling of how suitable a certain configuration is; the steeper slope the better.

Future work includes more thorough analysis in more scenarios with varying numbers of sensor nodes and network sizes, as well as evaluating alternative algorithms for cluster head selection. A comparison between the minimum separation distance algorithm and the BCDCP algorithm is also to be considered in the future.

Bibliography

[1] G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[2] M. Gerla, T.J. Kwon, and G. Pei. On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering. *Procedings of IEEE WCNC 2000*, Sep 2000.

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Maui, Hawaii, Jan 2000. In Proceedings of the 33rd International Conference on System Sciences (HICSS '00).

[4] G. Pei and C. Chien. Low Power TDMA in Large Wireless Sensor Networks. *Military Communications Conference*, 1:347–351, 2001.

[5] J. Neander, E. Hansen, M. Nolin, and M. Björkman. Asymmetric Multihop Communication in Large Sensor Networks. *International Symposium on Wireless Pervasive Computing 2006, ISWPC*, Jan 2006.

[6] W. Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts institute of technology, June 2000.

[7] T. Maruta and H. Ishibuchi. Performance Evaluation of Genetic Algorithms for Flowshop Scheduling Problems. *Proceedings of the 1st IEEE Conference on Evolutionary computation*, 2:812–817, June 1994.

[8] S. D. Muraganathan, D. C. F. Ma, R. I Bhasin, and A. O. Fapojuwo. A centralized energy-efficient routing protocol for wireless sensor networks. *Communications Magazine, IEEE*, 43(3):8–13, March 2005.

[9] NS 2, Network Simulator. World Wide Web, `http://www.isi.edu/nsnam/ns/ns-build.html`, 2004.

Chapter 9

Paper C: A Study of Maximum Lifetime Routing in Sparse Sensor Networks

Ewa Hansen, Mikael Nolin and Mats Björkman
To appear in proceedings of the International Workshop on Wireless Ad Hoc and Mesh Networks 2008, WAMN, Barcelona, Spain, March 2008.

Abstract

A major issue in wireless sensor networks is to prolong network lifetime by efficient energy management. In this paper we present an initial study of maximum lifetime routing in sparse sensor networks. We have studied simulations of how different heuristic routing algorithms influence the energy consumption of individual sensor nodes, and thus the functional lifetime of a sparse sensor network. The functional lifetime of the sensor network can be either until the first node has run out of energy or until a certain threshold of nodes has demised. We have also compared the maximum lifetime of the heuristic algorithms to the maximum lifetime of an optimal routing solution. Our simulations with non-aggregated data indicates that using one of the presented heuristic routing algorithms are not enough to find a near optimal routing. Our study is made in the AROS framework.

9.1 Introduction

Wireless sensor networks are rapidly becoming common in application areas where information from many sensors is to be collected and acted upon. Using wireless sensor networks adds flexibility to the network, and the cost of cabling can be avoided. One major issue in sensor networks is that wireless nodes most often obtain energy from a local battery. Since this limits the amount of energy available to the node, it affects the lifetime of the node and thus also the functional lifetime of the sensor network. In many application scenarios, replacement or recharging of power resources is costly or even impossible. Energy efficiency thus becomes a major issue in wireless sensor networks.

In the AROS project [1], an asymmetric routing has been developed, where communication links do not need to be capable of duplex communication. Instead, high-power nodes may transmit directly over longer distances, where low-power nodes would transmit using several shorter hops to cover the same distance. The applications are envisioned in areas where wired infrastructure is available, and where the degree of node mobility is low. Two examples of such areas are industrial and hospital environments.

In this paper we present an initial study of maximum lifetime routing in sparse sensor networks. We have studied simulations of how different heuristic routing algorithms influence the energy consumption of individual sensor nodes, and thus the functional lifetime of a sparse sensor network. The functional lifetime of the sensor network can be either until the first node has run out of energy, or until a certain threshold of nodes have demised, i.e. have no more energy to use.
We have also compared the maximum lifetime of the heuristic algorithms to the maximum lifetime of an optimal routing solution. In our ongoing project, we plan to investigate whether there is one single heuristic algorithm that suits our type of networks best, or if several heuristics need to be combined. The ongoing work also includes the balancing of energy consumption in the network in order to avoid hotspots draining individual nodes.

The rest of this paper is outlined as follows: in Section 9.2 we introduce the AROS framework. In Section 9.3 we describe some related work. Section 9.4 presents the different heuristic algorithms used in our simulations and in Section 9.5 the simulation setup is described. The results is presented in Section 9.6, and finally in Section 9.7 our conclusions are presented together with some future work.

9.2 The AROS architecture

The studies presented in this paper are made in the AROS framework [1], and we have therefore focused on sensor networks with infrastructure support. One (or more) of the network nodes is thus connected to the outside world, and to a power source such that power consumption is not an issue for that node. Such a node we call a *Base Station* (BS). The Base Station acts as a receiver (sink) of all sensor information produced by the network sensor nodes.

The AROS architecture [1] is based on cluster groups using base stations with "unlimited" energy and "enough" bandwidth in the backbone network. AROS uses a centralized approach to TDMA-based scheduling where resource-adequate Base Stations have global knowledge of the network and perform all calculations necessary to evaluate routes and schedules, thus relieving sensor nodes from the energy-consuming task of executing complex distributed decision algorithms. The sensor nodes periodically receive updated routing and scheduling information from the Base Station.

AROS divides the time in the network into *rounds*. One round is the time during which all sensor nodes (that wish to send data) send their data to the Base Station. After each round, routes and schedules are recalculated by the Base Station and distributed to the sensor nodes. This is done in order to minimize the risk of one sensor node forwarding too much data and therefore running out of power much earlier than the other sensor nodes in the network.

The communication between the Base Station and the sensor nodes is asymmetric, i.e. the Base Station can communicate directly with all sensor nodes, but the sensor nodes might have to communicate with the Base Station through other nodes, i.e. multihop.

9.3 Related Work

A lot of work has been done in the areas of energy efficient routing and power aware routing, e.g. [2, 3, 4, 5, 6] to name a few.

Singh *et* al. [7] presents the PAMAS protocol which is a MAC layer protocol that turns off the radio when the node is not transmitting or cannot receive packets. This protocol saves 40-70% of battery power according to [7]. The paper also includes several power aware metrics that are used to construct energy-efficient routes e.g. *Minimize Energy consumed/packet* and *Maximize Time to Network Partition*.

Li *et* al. [8] presents the *max-min* $z\mathrm{P}_{min}$ algorithm, which combines the benefit of selecting path with both the minimum power consumption and the path that maximizes the minimal residual power in the nodes of the network. An important factor in the *max-min* $z\mathrm{P}_{min}$ algorithm is the parameter z that tries to find a balance between the maximum minimum residual power path and the minimal power consumption path, but it seems that it is not so easy to find the optimal value of z. According to [8], the algorithm requires knowledge about each node in the network which can be a problem when implementing the algorithm in large networks. To solve this problem they propose a zone-based routing that relies on *max-min* $z\mathrm{P}_{min}$ but is scalable. In zone-based routing the network is divided into smaller zones, and each zone has only control over how to route the messages within its own zone. A global path across zones is also computed.

Chang *et* al. [2] presents a flow augmentation algorithm (FA) which is a shortest cost path routing where the link cost is a combination of transmission and reception energy consumption and the residual energy level at the two end nodes. The objective in [2] is to find the best link cost function which leads to the maximization of the system lifetime. When there is plenty of residual energy in the nodes, the energy cost term is emphasized, but when the node has less residual energy, the residual energy term has greater impact,i.e. is given more weight in the cost function.

Shah *et* al. proposes in [9] a scheme called energy aware routing that uses sub-optimal communication paths occasionally. The basic idea behind the scheme is to increase the survivability of the network by sometimes communicating through a sub-optimal path. They use a set of good paths and choose one of them, based on some probabilistic function. This means that instead of using one single communication path, different communication paths will be chosen at different times, thus any single communication path will not suffer from energy exhaustion.

9.4 Heuristic algorithms

As mentioned above, we want to maximize the functional lifetime of our network. Depending on the application and the amount of redundancy in the network, the functional lifetime can range from the time when the first node demises (in the case of no redundancy) to the time when all nodes have demised (in the case of full redundancy).

Our envisioned applications (industrial, hospital, domestic) are not likely to provide full redundancy. That means, the lifetime of individual nodes become more important. For a network with no redundancy, the optimal algorithm should keep all nodes alive as long as possible, i.e. until all nodes run out of energy at the same time. For a network with some degree of redundancy, some nodes can be allowed to run out of energy early, if that prolongs the lifetime of the rest of the nodes.

An optimal routing strategy should be able to construct a new routing scheme for every round in the network. Since our envisioned applications are not fully static (albeit slow to change), the new routing schemes must be constructed within limited time frames. Hence, it is not possible to make an exhaustive off-line scheduling of the network for its entire lifetime. Rather, the time available to construct the new schedule is likely to be in the order of a few seconds (or maybe even less). This implies that we must use efficient heuristics in order to meet the timing demands of the applications.

In our first approach to find such heuristics, we have investigated the relative efficiency of a number of heuristic algorithms. We want to find out if there is one single heuristic that suits our type of networks best. Should this not be the case, we want, in future studies, find under what circumstances the different algorithms are most efficient. If we can find good heuristics for when to change algorithm, we could in this case be more efficient than always using one single algorithm.

9.4.1 The algorithms studied

In order to find the most power efficient routes in our network, we have studied a number of simple heuristic algorithms that can be used to approximate the optimal routes. In this section we describe the algorithms we have studied.

During one *round*, all nodes send their sensed information/data *once* to the Base Station. The information is either sent directly to the Base Station, or through other nodes. When information from all nodes has been sent to the Base Station, a new routing scheme is made and a new round begins.

Minimum total energy consumption, MTEC

In the first algorithm, MTEC, we want to minimize the total energy consumption for the whole network, as in equation 9.1. In equation 9.1, e_i is the energy consumption for node i when sending to the Base Station and n is the number of nodes.

$$Min \sum_{i=1}^{n} e_i \tag{9.1}$$

The rationale behind this algorithm is that a smaller total energy consumption in the current round means that more energy will be left to coming rounds, i.e. the network as a whole will live longer. The balance between the energy consumption of individual nodes is however not considered in this algorithm.

Minimum squared energy consumption, MSEC

The second algorithm, MSEC, is based on the consideration that one node can be very heavily loaded, but the total energy consumption can still be the lowest. In this algorithm, we square the energy consumption of each individual node before we sum the energy consumption, as in equation 9.2.

$$Min \sum_{i=1}^{n} (e_i)^2 \tag{9.2}$$

The rationale behind this algorithm is that routes where one node is heavily loaded will get a higher sum and thus be less likely to be chosen as the best route. Hence, we will get more equally loaded nodes than in the first algorithm, equation 9.1, while still choosing a route with a low total energy consumption.

Minimal maximum individual energy consumption, MMIEC

In our third algorithm, MMIEC, we minimize the maximum energy consumption for a single node. This is shown in equation 9.3, where e_{max} is the maximum energy consumption for a single node in the chosen route.

$$Min(e_{max}) \tag{9.3}$$

The rationale behind this algorithm is that if we can minimize the maximum energy consumed by one node, we can prolong the lifetime for the node that consumes the most energy for a given route, and thereby prolong the lifetime of the whole network. One drawback can be that if all nodes consume almost the same amount of energy the network may demise quickly.

Minimal difference in energy consumption, MDEC

Our fourth algorithm, MDEC, makes the difference in energy consumption between the most consuming and the least consuming node as small as possible. This is shown in equation 9.4, where e_{max} is the maximum energy consumption for a single node and e_{min} is the minimum energy consumption for another single node.

$$Min(e_{max} - e_{min}) \tag{9.4}$$

The rationale behind this algorithm is that this algorithm makes the average energy consumption approximately equal between the nodes. This approach can however be less efficient if all nodes consume a lot of energy. In this case the difference between the nodes' energy consumption can be small but the energy consumption for each individual node might be high. This would lead to shorter lifetime for the network.

Maximum squared remaining energy, MSRE

In the fifth algorithm, MSRE, we have studied the remaining energy of the nodes, taking the maximal sum of the squares of the remaining energy. This is shown in equation 9.5, where e_{left} is the remaining energy for a single node.

$$Max \sum_{i=1}^{n} (e_i^{left})^2 \tag{9.5}$$

This algorithm tries to maximize the remaining energy of the system. Since the square of the remaining energies are used in the sum, the algorithm will favor routes where one (or more) nodes have much energy left. For networks where the functional lifetime of the network continues until all nodes have demised, this can be beneficial. However, since the algorithm favors energy unbalance in the network, the first node (or nodes) is likely to demise earlier than when using algorithms that favor energy balance.

Maximal minimum individual remaining energy, MMIRE

The sixth algorithm, MMIRE, maximizes the minimum energy left for a single node. This is shown in equation 9.6.

The energy left after a chosen round is calculated in advance and the most exposed node, i.e. it has the lowest energy left, is maximized, this to not expose one single node more than needed.

$$Max(e_{min}^{left}) \tag{9.6}$$

The rationale behind this algorithm is that it makes the most exposed node during one round hopefully less loaded during the next rounds, thus spreading the energy consumption more evenly over the nodes. One drawback with this approach is that only one node is under consideration.

9.5 Simulation setup

For the simulations presented in this paper we have implemented a routing system and a simulator.

We have made simulations with 100 randomly generated sensor networks. The network area was 400x400 m^2 and the number of nodes randomly spread across the network was 5. These nodes can be considered as either ordinary sensor nodes or cluster head nodes in a cluster-based sensor network, e.g. as in the AROS project [1]. The reason for using a small amount of nodes is that we want to be able to compare results from our heuristic routing algorithms to results using an optimal routing solution. To simulate the optimal routing is too resource-consuming to be feasible to calculate for larger numbers of nodes. To be able to find the optimal route, we have made a complete search among all possible routes, and the most energy efficient[1] result is found.

When calculating the energy consumption of the sensor node radio transmitter, we have used the same equation as in [1, 10, 11]. When sending a message a distance up to 87 meters, we have used $\epsilon_{friss-amp} = 10pJ/bit/m^2$, and when sending a distance of more than 87 meters we have used $\epsilon_{two-ray-amp} = 0.0013pJ/bit/m^4$. The radio electronics consume $E_{elec} = 50nJ/bit$. The equation for calculating the total amount of energy consumed when sending a message of b bits a distance of d meters is then:

$$E_{Tx} = \left\{ \begin{array}{ll} b * E_{elec} + b * \epsilon_{friss-amp} * d^2 & : d < 87m \\ b * E_{elec} + b * \epsilon_{two-ray-amp} * d^4 & : d \geq 87m \end{array} \right. \tag{9.7}$$

The amount of energy used by a sensor node radio receiver when receiving a message is:

$$E_{Rx} = b * E_{elec} \tag{9.8}$$

---

[1] Most number of rounds.

Table 9.1: Non-Aggregated data

	MTEC		MMIEC		MDEC	
	$\bar{a}$	σ	$\bar{a}$	σ	$\bar{a}$	σ
5 nodes	13,11	13,17	15,93	14,18	14,83	12,86
4 nodes	7,59	8,76	5,3	11,18	4,54	10,33
3 nodes	6,88	11,25	6,39	12,04	6,10	13,51
2 nodes	13,69	29,11	11,26	30,77	11,26	33,88
Total	**41,27**	**16,36**	**39,38**	**17,70**	**36,73**	**18,57**
	MSRE		MMIRE		MSEC	
	$\bar{a}$	σ	$\bar{a}$	σ	$\bar{a}$	σ
5 nodes	12,27	12,38	2,07	1,60	15,27	13,95
4 nodes	7,49	8,72	3,28	3,68	6,16	10,96
3 nodes	7,09	11,70	5,59	6,30	6,56	11,94
2 nodes	14,79	30,92	21,35	32,59	12,62	32,86
Total	**41,64**	**16,97**	**32,29**	**16,75**	**40,61**	**18,27**

In our simulations, each node starts with an energy of 0,1 mJ[2]. All nodes consume energy when transmitting and receiving data packets. When not transmitting or receiving, the nodes are in sleep mode and are assumed to use very little energy. In this paper this energy is assumed negligible.

We have performed simulations with both aggregation and non-aggregation of data, where aggregation means that a downstream node can aggregate two (or more) messages of size N, bound for the same destination, into one single message of size N. This enables us to study if there are differences between aggregation and non-aggregation with respect to what algorithms perform best in our scenarios.

9.6 Results

In this section, we present some results from our studies. We have made an initial study of maximum lifetime routing in sparse sensor networks. We have studied simulations of how different heuristic routing algorithms influence the energy consumption in individual sensor nodes, and thus the functional lifetime of a sparse sensor network. We have also compared the maximum lifetime of

---

[2] The reason for the small amount of initial energy is due to the execution time of the simulation.

the heuristic algorithms to the maximum lifetime of an optimal routing solution.

The results are from simulations with non-aggregated data and from simulations with aggregated data. We have calculated the energy consumption E_{Tx}, using equation 9.7, when sending and E_{Rx}, using equation 9.8, when receiving. All nodes that receives data consumes E_{Rx} for each message it receives. When aggregating data a node uses E_{Tx} for the one (aggregated) message it sends, and when not aggregating data it uses E_{Tx} for each message it forwards.

We have also compared the results from the different algorithms with an optimal routing solution. The optimal routing solution in this paper is simulated the same way as the other algorithms, but instead of running one of the heuristic algorithms, a complete search tree is computed and the most energy efficient[3] result is found.

9.6.1 Results of heuristic algorithms

In tables 9.1 and 9.2, we can see results from the six algorithms described in section 9.4. The average number of rounds ($\bar{a}$) and the standard deviation (σ) are calculated for all the algorithms. We also show all the separate numbers of rounds, from when the first node runs out of energy until it is only one node left in the network. i.e. 5 nodes = number of rounds with all nodes alive, 4 nodes = number of rounds with one node demised, and so on. *Total* is the total number of rounds until all nodes have run out of energy. We have, in this paper, concentrated on two different functional lifetimes, as mentioned in Section 9.4. The two functional lifetimes are; until the first node demises and until all nodes have demised.

Non-aggregated data

When looking at the results from our simulations with non-aggregated data, found in table 9.1, we can see that the average number of rounds until the networks have demised differs a bit among the algorithms, although in most cases not much. When choosing route using MSRE (equ. (9.5)), we can see that this results in the most energy-efficient routing, but using MTEC (equ. (9.1)) is almost as good. When choosing MMIRE, (equ. (9.6)) we can see that this approach is not quite as good as the other approaches. MMIRE is not good at all if we want to maximize network lifetime until the first node demises. We

---

[3] Most number of rounds.

can see that the other algorithms run approximately 6 to 8 times more rounds before the first node demises.

Another point worth noting is that MDEC has the shortest total network lifetime, if not including MMIRE. This is not surprising, since, as noted in Section 9.4, MDEC will even out energy differences, but does not consider the total energy consumption. An indication of this property is also that the lifetime until the first node demises is relatively good for MDEC, since MDEC tries to balance energy consumption as much as possible, thus keeping all nodes alive for a relatively long period.

MSRE is quite the opposite to MDEC. MSRE has bad results for the number of rounds until the first node demises, but has the longest total lifetime of all algorithms. This is consistent with the discussion in Section 9.4, MSRE favors one (of a few) nodes with much energy left, and this is likely to lead to the early demise of one of the other nodes.

Aggregated data

When looking at the results from our simulations with aggregation of data, found in table 9.2, the differences among the algorithms are not big, although there are some differences. In these simulations one of the algorithms is again different from the others, MMIRE (equ. (9.6)). When aggregating data, the other algorithms runs approximately 10 times more rounds, compared to MMIRE, before the first node demises.

The conclusions from these comparisons are that several of the heuristic algorithms exhibit a similar behavior, when looking at the mean values and standard deviations of the same 100 generated networks. Also, it is clear that the MMIRE algorithm is not as good as the other heuristic algorithms.

9.6.2 The algorithms compared to optimal results

When simulating the optimal routing solution, we selected one of the most energy-consuming networks among the 100 randomly generated networks, and compared the result with our heuristic algorithm results. The reason for choosing one of the most energy-consuming networks was due to the execution time of the optimal solution. The cost of finding the optimal solution is exponential to the number of rounds, so only networks with small numbers of rounds are feasible to find the optimal solution for.

Table 9.2: Aggregated data

	MTEC		MMIEC		MDEC	
	$\bar{a}$	σ	$\bar{a}$	σ	$\bar{a}$	σ
5 nodes	37,75	35,00	37,8	35,02	37,67	35,04
4 nodes	20,96	28,54	20,9	28,65	19,79	29,26
3 nodes	19,79	38,60	19,56	38,66	15,98	33,03
2 nodes	29,77	68,89	29,54	68,92	28,4	71,91
Total	**108,27**	**42,45**	**107,8**	**42,48**	**101,84**	**42,61**
	MSRE		MMIRE		MSEC	
	$\bar{a}$	σ	$\bar{a}$	σ	$\bar{a}$	σ
5 nodes	37,41	34,56	3,73	3,39	37,79	35,02
4 nodes	21,01	29,08	4,66	4,82	20,99	28,54
3 nodes	19,72	38,57	8,55	10,67	19,82	38,64
2 nodes	29,57	69,21	26,48	44,57	29,68	68,88
Total	**107,71**	**42,51**	**43,42**	**22,59**	**108,28**	**42,45**

Non-aggregated data

When comparing the non-aggregated results from the heuristic algorithms with the optimal solution for non-aggregated data, the differences are more significant when comparing the number of rounds until all nodes have demised. None of the heuristic algorithms could match the optimal solution of a total of 9 rounds. The two heuristic algorithms that managed best were MTEC and MSRE with 7 rounds. MMIEC and MMIRE managed 6 rounds and MDEC and MSEC only managed 5 rounds before all nodes had demised.

Comparing the number of rounds until one node had demised resulted in 3 rounds for the optimal solution, MMIEC, MDEC, and MSEC. MTEC managed 2 rounds and MSRE and MMIRE only 1 round. Again MMIRE, as mentioned above, is not as good as the other algorithms.

The conclusions from this comparison are first of all that for non-aggregated data, the heuristic algorithms were far from optimal even for a network that only survived 9 rounds. Also, there are clear differences between the heuristic algorithms when examining one single network. Finally, it is clear that the MMIRE algorithm is not a good algorithm.

Aggregated data

When comparing the aggregated data simulations to the optimal routing solution for aggregated data, the differences are very small or none. (We only compared the total number of rounds, and the number of rounds until one node had demised.) The total number of rounds for the optimal solution and for four of the heuristic algorithms was 13 rounds. The algorithms that were different were MDEC and MMIRE, which had fewer rounds, 10 and 6 respectively.

When comparing the heuristic algorithms to the optimal solution until one node had demised, there was only one algorithm, MMIRE, that showed fewer rounds, 2, than the optimal solution. All the other algorithms showed the same number of rounds, 4, as the optimal solution. (As mentioned earlier, the MMIRE algorithm is not as good as the other algorithms.)

9.7 Conclusions and Future Work

In this paper we have made an initial study of maximum lifetime routing in sparse sensor networks. We have studied simulations of how different heuristic routing algorithms influence the energy consumption in individual sensor nodes, and thus the functional lifetime of a sparse sensor network. We have also compared the maximum lifetime of the heuristic algorithms to the maximum lifetime of an optimal routing solution.

We have performed simulations with 100 randomly generated sensor networks where the network area was 400x400 m^2 and the number of nodes randomly spread across the network was 5. The simulations were made with both aggregation and non-aggregation of data, and a comparison with an optimal routing solution was also done.

When looking at the simulation results with aggregated data we can see that it is not a big difference among the heuristic algorithms. The algorithms MSEC and MTEC are the two heuristic algorithms that show the best results. The heuristic algorithm that shows the worst results is clearly MMIRE, see table 9.2.

When comparing these heuristic algorithms to the optimal routing solution (one of the most energy consuming network setups), the differences are very small or none. The total number of rounds for the optimal solution and for four of the heuristic algorithms was 13 rounds. The algorithms that were different were MDEC and MMIRE, which had fewer rounds, 10 and 6 respectively.

When looking at the simulation results with non-aggregated data, the differences among the heuristic algorithms were slightly bigger. If only looking at the total number of rounds until all nodes have demised, MSRE, MTEC, MSEC and MMIEC were the four heuristic algorithms that performed best (when comparing both the average number of rounds and the standard deviation). When comparing the number of rounds until one node had demised, MSEC and MDEC were slightly better than the others. Looking at MMIRE, we can see that this heuristic algorithm is not good at all if we want to maximize network life time until the first node demises.

Comparing to the optimal routing solution, the differences are more significant when comparing the total number of rounds. None of the heuristic algorithms could match the number of rounds for the optimal solution. The two heuristic algorithms that managed best were MTEC and MSRE.

The conclusions of these simulations are that when aggregating data, the choice of heuristic algorithm is not as significant as when not aggregating data. Some differences have been identified and one of them is that MMIRE is not a good heuristic algorithm.

Our simulations with non-aggregated data indicates that using one of the presented heuristic routing algorithms are not enough to find a near optimal routing, hence it is possible that several different heuristic algorithms need to be combined to find a near optimal routing solution.

In the future we will continue our work to prolong network lifetime e.g. until the first node demises (in sparse networks) or until some threshold of nodes have demised (in more densely populated networks). The initial studies in this paper is the beginning of ongoing work where we plan to investigate how we can combine these heuristic algorithms to be able to find a near optimal routing solution. We will investigate when to change heuristic and what heuristic that is most suitable in different situations. We will also investigate for what kinds of network setups different heuristic algorithms are most suitable, e.g. for what kind of network setup is MMIES most suitable? In future work we will also try to find a near optimal routing solution by e.g. weighting each link so that no node drains its energy faster than the other nodes, i.e. avoiding hotspots.

Bibliography

[1] J. Neander, E. Hansen, M. Nolin, and M. Björkman. Asymmetric Multihop Communication in Large Sensor Networks. *International Symposium on Wireless Pervasive Computing 2006, ISWPC*, Jan 2006.

[2] J. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(4):609–619, 2004.

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Maui, Hawaii, Jan 2000. In Proceedings of the 33rd International Conference on System Sciences (HICSS '00).

[4] S. D. Muraganathan, D. C. F. Ma, R. I Bhasin, and A. O. Fapojuwo. A centralized energy-efficient routing protocol for wireless sensor networks. *Communications Magazine, IEEE*, 43(3):8–13, March 2005.

[5] S. Ming Tseng and R. Pandey. A Hierarchical Routing Protocol for Networks of Heterogeneous Sensors. Goteborg, August 2004. 10th International Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA).

[6] M. Younis, M. Youssef, and K. Arisha. Energy-aware management for cluster-based sensor networks. *Computer Networks*, 43:649–668, Dec 2003.

[7] S. Singh, M. Woo, and C. S. Raghavendra. Power-Aware Routing in Mobile Ad Hoc Networks. In *Mobile Computing and Networking*, pages 181–190, 1998.

[8] Q. Li, J. Aslam, and D. Rus. Online power-aware routing in wireless Adhoc networks. In Proceedings of the 7th annual international conference on Mobile computing and networking, 2001.

[9] R. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. *In Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, March 2002.

[10] E. Hansen, J. Neander, M. Nolin, and M. Björkman. Energy-Efficient Cluster Formation for Large Sensor Networks using a Minimum Separation Distance. Lipari, Italy, Jun 2006. In Proceedings of The Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2006).

[11] W. Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts institute of technology, June 2000.