Mälardalen University Press Licentiate Thesis
No.66

Using Existing Infrastructure as Support for Wireless Sensor Networks

Jonas Neander

June 2006

CUGS
National Graduate School of Computer Science

MÄLARDALEN UNIVERSITY

Mälardalen Real-Time research Centre (MRTC)
Department of Computer Science and Electronics
Mälardalen University

ISSN 1651-9256
ISBN 91-85485-17-9
Printed by Arkitektkopia, Västerås, Sweden
Distribution: Mälardalen University Press

Abstract

Recent advancements in electronic design, such as low-power circuits, energy efficient wireless communication, and improved energy supply, has enabled the vision of wireless sensor networks to become a reality. Wireless sensor networks typically consist of hundreds up to thousands of collaborating low-cost, battery-driven and wireless sensor nodes with scarce resources. The wireless sensor nodes are typical small physical entities, and usually small as a matchbox but can in extreme cases be no larger than a cubic millimeter.

In this thesis we present an architecture called AROS that uses existing infrastructure to aid in the management of wireless sensor networks. As an example, the existing infrastructure could be situated in hospitals or industrial buildings. The existing infrastructure can aid in prolonging the lifetime of the wireless sensor network by having “unlimited” energy, long range radio capacity, and high-speed computers. We enable prolonged lifetime by centralizing some of the energy consuming administrative functionality of wireless sensor networks.

We show, by simulations, that the AROS architecture is able to prolong the lifetime of the sensor nodes. AROS is compared to a well known cluster based architecture, LEACH. The comparisons show that AROS with static configuration performs at least as well as LEACH in small wireless sensor networks in the size 100x100m, and up to 97 % better in long distance wireless sensor networks in the size of 400x400m. We show that AROS still has got 88 % of its sensor nodes alive when LEACHs’ network demises.

In our simulations we have also studied how dynamic network clustering in AROS, using a TDMA scheduler and non-mobile wireless sensor nodes, affects the amount of data received by a base station. We show that AROS is better than LEACH-C in collecting data to the base station with the same total amount of energy for long distance networks and that AROS performs as well or better than LEACH-C in small wireless sensor networks.

Swedish summary - Svensk sammanfattning

Denna avhandling handlar om hur befintliga datorinfrastrukturer i t.ex. sjukhus och industrier kan avlasta sensornätverk med energikrävande uppgifter. Vi har forskat på olika aspekter som gör det möjligt att förlänga livslängden på dessa sensornätverk. Avhandlingen presenterar en ny plattform för sensornätverk tillsammans med inledande simuleringar som påvisar att vår plattform ökar livslängden på dessa typer av nätverk.

Generella sensornätverk är uppbyggda av tätt grupperade, trådlösa, batteridrivna datorer som kan vara så små som en kubikmillimeter. Datorerna kallas för sensorer eller sensornoder eftersom de har en eller flera inbyggda sensorer som känner av sin omgivning. En sensor har till uppgift att samla information från sin omgivning, t.ex. temperatur, fuktighet, vibrationer, hjärtslag eller bilder. Sensorerna skickar sedan informationen till en insamlingsstation någonstans i nätverket.

I de typer av tillämpningar vi tittar på är det viktigt att minimera energiförbrukningen, så att man maximerar livslängden på sensornätverket. Avhandlingen presenterar en lösning där befintlig datorinfrastruktur fungerar som hjälpdatorer/avlastare till sensornätverken. Hjälpdatorerna, eller basstationerna som vi kallar dem i avhandlingen, hanterar energikrävande uppgifter som t.ex. vilken sensor som ska kommunicera med vem samt vid vilken tidpunkt etc. Då kan sensorerna i nätverket fokusera på att utföra sina egna uppgifter tills dess att basstationen säger att uppgifterna ändrats.

Simuleringar visar att vår plattform kan skicka upp till 97 % mera information till basstationen än en jämförbar plattform med samma energimängd. 88 % av våra sensorer är fortfarande vid liv när den andra plattformens sensorer förbrukat all sin energi.

Ett exempel på hur dessa typer av nätverk kan användas är att övervaka patienters hälsa och kondition i sjukhus eller sjukhem. Patienter behöver inte ha en fast sängplats där en viss typ av medicinskt övervakningsinstrument finns tillgänglig utan kan placeras där det finns en ledig sängplats. Via trådlös kommunikation skickar sensorerna sedan hälsoinformation som t.ex. hjärtfrekvens och blodtryck till en basstation som i sin tur skickar vidare till ett centralt övervakningsinstrument någonstans på sjukhuset. Övervakningsinstrumentet behandlar informationen och larmar personal med rätt kompetens vid behov. Larmet kan skickas till en mobiltelefon eller en liten handdator som personalen alltid bär med sig. Med larmet skickas även information om var patienten befinner sig och all nödvändig data för att personalen snabbt ska kunna ställa en första diagnos. På detta sätt kan man spara in på antalet specialbyggda sängplatser och slippa dyrbara installationer av medicintekniska utrustningar knutna till en sängplats.

Till Josephine och Julia
Jag älskar er över allt annat

Preface

This research work was funded in part by CUGS (the National Graduate School in Computer Science, Sweden).

First, I would like to thank my supervisors, Professor Mats Björkman, Associate Professor Mikael Nolin and Dr. Jukka Mäki-Turja for their excellent supervision during these years. Thank you all, a lot! Secondly, I would like to thank my closest colleagues Ewa Hansen and Andreas Johnsson for the fruitful discussions and laughs over the years. Thank you both. Thanks to Javier and Denny for answering all my stupid radio questions. I think that I still remember some of the very thorough answers! A big thanks to all my colleagues and friends at the department. It has been a pleasure to work with you and sharing spare time together with you guys. I also like to thank one former and perhaps future colleague. A guy who I've become very found of. Thank you for all the laughs and jokes Dr. Lennis.

This thesis is dedicated to my wonderful daughters Josephine and Julia. I started to study again for their sake a couple of years ago, eleven years ago actually. The only problem seems to be that I don't know when to quit. I never intended to study this far/long. I love you both so very much.

I like to thank my mother, Sonya, my girlfriend Ewa and her son Marcus and my three brothers, Jerry, Robin and Patrik for their support and love. I also would like to thank my "new" family Stefan, Maria, Inge and Elin for unforgettable moments in life.

People who knows me know that I'm very fond of playing golf. I like to thank my girlfriend Ewa for all the laughs and tears we shared on the golf course. I'm a lucky man having such as golf fanatic girlfriend as you. I would also like to take the opportunity to thank some other people that I have had the pleasure of playing a round or two with. Jukka the hard hitting "joppe", you are better in supervising me at work than supervising me at the golf course, that's for sure. Dr. Dag, plays like a newbie but thinks he's a pro on the PGA tour. Keep up the good work, you are almost there! Inge, soon capable of opening his own golf store at home. How can you manage to never get angry or disappointed when playing golf? I really could use some of your temperament on the golf course. Last but not least Manne. Too bad you are moving out of town, I really need someone that easy to beat on the golf course.

I also would like to thank all my friends. Jeppe, Jonas, Peter & Tina, Marie, Bea, Johan A, Cribbe, Krasse, Blom & Ruth, Joel & Rebecca, Lotta, Sofia, Sara, Caroline, Daniel, Nolte, Pettsson, Radu, Åkerholm, Larisa, Damir, Fredriksson, Peter W and Esa. Thank you all for sharing my life.

Jonas Neander
Västerås, May 30, 2006

Contents

List of publications

Publications included in the licentiate thesis

Paper A: *Using Existing Infrastructure as Proxy Support for Sensor Networks*, Jonas Neander, Mikael Nolin, Mats Björkman, In 16th EUROMICRO Conference on Real-Time Systems (ECRTS 04), Work in progress, Catania, Italy, June, 2004.

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

Paper C: *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.

I have been the main driving author of these papers and I wrote most of the text for the papers. In Paper B, my co-worker and co-writer Ewa Hansen and I implemented AROS and we performed the simulations in NS-2.

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, The Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2006).

- *Energy-Efficient Cluster Formation for Large Sensor Networks using a Minimum Separation Distance*, Ewa Hansen, Jonas Neander, Mikael Nolin and Mats Björkman, The Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2006).

- *Proxy Support for Sensor Networks Using Existing Infrastructure*, Jonas Neander, Mikael Nolin, Mats Björkman, Medicinteknikdagarna 2005, Södertälje, Sweden, September, 2005. NOTE: same paper as published in SNCNW2004.

- *Proxy Support for Sensor Networks Using Existing Infrastructure*, Jonas Neander, Mikael Nolin, Mats Björkman, 2nd Swedish National Computer Networking Workshop, SNCNW2004, Karlstad, Sweden, November, 2004.

- *Introducing Temporal Analyzability Late in the Lifecycle of Complex Real-Time Systems*, Anders Wall, Johan Andersson, Jonas Neander, Christer Norström, Martin Lembke, In proceedings of RTCSA 03, February, 2003.

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.

- *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.

- *An Asymmetric Proxy Backbone Architecture for Sensor Nodes*, Jonas Neander, Mikael Nolin, Mats Björkman, MRTC report ISSN 1404-3041 ISRN MDH-MRTC-158/2004-1-SE, Mälardalen Real-Time Research Centre, Mälardalen University, April, 2004.

I

Thesis

Chapter 1

Introduction

In this thesis we investigate how existing infrastructure can be utilized to prolong the lifetime of wireless sensor nodes in wireless sensor networks. We assume that the sensor nodes are not necessarily able to communicate directly with the infrastructure nodes. Existing infrastructure can be situated in, e.g., hospitals and industrial buildings and can be used as support for the wireless sensor network.

Generally, a wireless sensor network consists of densely deployed wireless sensor nodes running on batteries. The wireless sensor nodes are typical small as a matchbox but can be small as a cubic millimeter. The wireless sensor nodes collaborate in a dense network that can consist of hundreds up to thousands of wireless sensor nodes. The wireless sensor nodes sense their own surroundings and report what they sense to a user or computer. A wireless sensor node can, e.g., sense the surrounded temperature, humidity, seismic activities/vibrations, heartbeats or can even take pictures [9]. The wireless sensor nodes often have short-range radios and often need to collaborate with each other in order to deliver the sensed data to the user or computer. Forwarding data from other wireless sensor nodes in the wireless sensor network is usually one of the most common forms of collaboration between the wireless sensor nodes.

Wireless sensor networks can be used, e.g., to monitor houses, fields, forests, lakes, oceans, merchandizes or processes in industries [1, 2, 3, 4, 5, 6, 7, 10, 16, 17, 18]. Further, the wireless sensor networks could be used for, e.g., surveillance of people (like patients in a hospital), in parking lots to monitor free parking spaces, to monitor animal life in forests or oceans [11, 16, 17, 20, 29, 26, 28, 33, 34]. The industry forecasts an explosive growth in the use of sensor network applications in industry in the near future [3]. The wireless sensor nodes in industry applications are intended to replace traditional wired sensors. We believe that these new applications, and applications considered too expensive before or even impossible, will in a near future will be a reality. The vision is that these wireless sensor networks should be cheap to build and maintain and the cost of a wireless sensor should not exceed one US dollar [27].

The most relevant metric in development of wireless networks is typically power. It has been shown with experimental measurements that the cost due to communication in wireless ad-hoc networks is at least two orders of magnitude higher than computation costs in terms of consumed power [24]. Having battery-driven wireless sensor nodes in the network leads to two important requirements:

- The limited lifetime of the wireless sensor nodes should be as long as possible, and
- the network should be robust and flexible enough to tolerate loss of, and replacements of, wireless sensor nodes.

Many areas where sensor networks could be deployed does already have existing infrastructure in terms of standard computers connected to each other in a network. Such areas include hospitals and industrial buildings. These networks often have a wired network together with wireless access points. Computers and other peripherals can connect to this network in order to access, e.g., data stored on a server, medical journal documents or home pages from the internet.

In this thesis we combine existing infrastructure, situated in for example hospitals and industrial buildings, with wireless sensor networks. We believe that we could increase the lifetime of the wireless sensor nodes when combining the wireless sensor networks with an existing infrastructure. The infrastructure could then help the wireless sensor nodes with energy consuming tasks. Traditional sensor nodes use a lot of energy when communicating to organize and maintain the wireless network and thus draining its energy capacity on administrative functionality instead of on productive sensing. If the infrastructure, having “unlimited” energy and high-speed computers, could take over these administrative duties, sensor networks could save energy and thus prolong the lifetime of the network.

1.1 Thesis outline

This thesis is organized into two parts. In the first part we give a short introduction and background to the research area. This part aims at introducing the readers not familiar with the wireless sensor network research area. The second part contains the scientific papers A, B and C.
Part I is organized as follows:

Chapter 1: This chapter briefly introduces the wireless sensor network area and describes what we mean by an existing infrastructure. It also motivates the research that has been accomplished in Part II.

Chapter 2: In this chapter we describe the wireless sensor network in more detail. We introduce the application areas where the sensor nodes are intended to operate, we describe how the wireless sensor networks differ from other wireless networks. We show two commonly used topologies and two possible ways to reduce the data traffic in the wireless network. Further, we show some challenges for the wireless networks and we end this chapter with a description of the wireless sensor node architecture and some relevant concepts.

Chapter 3: In Chapter 3 we define what we mean by existing infrastructure networks and state the problem formulation.

Chapter 4: The architecture of AROS is described in detail in this chapter together with some tradeoffs needed to be considered when designing a wireless sensor network.

Chapter 5: Related work is presented in this chapter and we describe how it is relates to AROS.

Chapter 6: In this chapter we summarize and present the contributions of the papers included in this thesis.

Chapter 7: We conclude Part I with a conclusion and point out some directions for future work.

Chapter 2

Wireless sensor networks

After briefly introducing wireless sensor networks in Chapter 1 we will in this chapter present the wireless sensor network area in more detail. We start by presenting some of the application areas for wireless sensor networks. For simplicity, we will throughout the thesis assume the sensor networks to be wireless unless otherwise explicitly stated.

2.1 Application areas

The application areas for sensor networks have a huge variety and have the potential to revolutionize information gathering and processing [11]. There are many possible application areas for sensor networks [10, 16], and in this section we discuss some of the areas and briefly describe how sensor networks are intended to operate in these areas.

- **Environmental:** Sensor networks could be situated on an island, monitoring the behavior of nesting birds in their own habitat without human interference/disturbance, as in the Great Duck Island project in Maine [20] monitoring the nesting Petrel. Or, the sensor nodes could be spread over a forest in order to, for example, detect possible forest fires or monitor an ongoing fire. A sensor network could be deployed at river sides monitoring the water level and alarm people living close to the river in case of flooding [2]. Agriculture applications could make use of sensor networks. A sensor network can for example monitor the dampness of the soil in order to irrigate more accurately. The marine can use sensor networks in order to monitor water currents or temperature changes in the oceans or rivers, e.g., CORIE [7], a pilot environmental observation and forecasting system (EOFS) for the Columbia River. Surveillance of areas in, e.g., military applications monitoring the movements of military units etc.

- **Health:** Monitoring elderly people with sensor networks, both at home and in geriatric care. Health monitoring includes hearth rate, blood oxygen saturation, temperature, people falling etc. In hospitals for instance, doctors' and nurses' health states could be monitored in order to prevent people to get ill due to stress. Patients in hospitals for instance, can be more mobile with sensor nodes since no cables need to be plugged in. Medical equipment can be equipped with sensor nodes in order to eliminate cables or to interact with a patient's sensor node telling the doctor what possible allergies the patient has or what medications the patient currently are using. Biomedical sensors for visually impaired in, e.g., the retina [29] connected to the optic nerve producing image signals to the brain.

- **Home:** Intelligent homes with sensor nodes monitoring things such as adjusting/optimizing the ventilation to multimedia applications. Optimizing the ventilation can reduce the energy consumption to heat/cool the house. Sensor nodes can monitor the refrigerator's contents and keep track of what is missing and need to be ordered or purchased. Sensor nodes can keep track of family members and in what room they are in. When a person for example leaves a room, the music or video stream can follow the person from one room to another automatically.

- **Industry:** The analysts in industry forecast an explosive growth in the use of sensor network applications in industry in the near future [3]. Sensor networks in industry will help to increase knowledge about the enterprize. The increased knowledge can be how the machinery works, production quality, where the merchandize is located, staff health monitoring, ventilation and temperature in buildings and surveillance.

  For example, merchandize in warehouses can be positioned with help from sensor nodes or the sensor nodes could be used to check the quality of provisions. The company wants to ship the oldest provisions before the younger to keep high quality of its stored provisions. The company does not want to condemn expired or stale provisions and they do not want to bring bad provisions to its customers. Therefore, keeping track of the provisions' freshness and expiration dates is important. Sensor nodes can be used in intruder surveillance in premises or on fences. Sensor nodes can be placed in machinery where cables are not feasible due to cost or limiting the flexibility/mobility. The sensor nodes could for instance be built into the concrete of bridges in areas with high risk of earthquakes, monitoring how seismic activities affect the integrity of the structure.

2.2 Sensor networks - A new family member

Sensor networks are a new family member in wireless networking family. Sensor networks differ from other family members, such as cellular networks and Mobile Ad hoc NETworks (MANETs) in the way the networks are designed and used. To show why these other networks are not suitable for sensor network applications we briefly describe some of them below.

2.2.1 Cellular networks

Cellular networks consist of non-mobile base stations and mobile nodes. The base stations have "unlimited" energy and are connected to each other by wire forming a backbone. Each base station covers a large network area up to 35 km and the base stations overlap some of each others network area. The mobile nodes communicate directly with the base station and the primary goal is to provide high Quality of Service (QoS) with enough communication speed and bandwidth. The energy consumption is of importance, albeit secondary, as the users recharge their cell phones when necessary.

2.2.2 Mobile ad hoc networks

The most common notion of a mobile ad hoc network is a network consisting of wireless mobile nodes formed without any help from, e.g., a central administration. The nodes in this network need to be prepared to act as routers, forwarding other nodes' data traffic. Mobile Ad hoc NETworks (MANETs) are designed as multihop[1] peer-to-peer networks with ten to hundreds of nodes with good energy capacity [31]. The nodes are often attached to a person, e.g., laptops or Personal Digital Assistants (PDAs) and they are mobile and

---

[1] In multihop networks the data travels, is forwarded, through several computers before reaching its final destination

equipped with wireless radio enabling the nodes in the network to cover areas up to hundreds of meters. The network is designed to transport traffic like voice, multimedia, mail, web surfing and file access. Good throughput with low delay under high mobility is of great importance in these networks. The tasks like, routing, organization of the nodes and mobility management is done to optimize the QoS in the network. The energy consumption is of secondary importance as batteries can be recharged or replaced when needed.

2.2.3 Sensor networks

Sensor networks are designed for unattended operations with hundreds up to thousands of scattered sensor nodes with limited energy capacity. The sensor nodes remain fairly stationary after deployment. While the data rate is high in Cellular networks and MANETs, it is, typically, low in sensor networks. Sensor networks often send a small amount of intermittent statistical data, around 1-100kb/s [26]. A common goal in sensor networks is to prolong the lifetime of the network at the expense of, e.g., data rate, delay time and QoS. Batteries can not always be replaced due to hostile, hazardous or remote environments, or it might not be cost-effective. Thus, low energy consumption is of great importance in order to prolong the lifetime of the network. Harsh environments increase node failures and the network needs to be fault tolerant and dynamically adaptable.

2.3 Common view of the sensor network topology

A common view of a sensor network topology is shown in Figure 2.1. The sensor nodes are scattered over an area represented by the cloud. It is a forwarding multihop sensor network, i.e., data travels, is forwarded, through several sensor nodes before reaching its destination. The sensor nodes use a predefined routing scheme to communicate with other sensor nodes, i.e., how the data should travel in the network. The destination of the data traffic is often represented by one or several sinks. A sink is the destination of the sensor nodes' data, usually a high-performance computer connected to a wired backbone and with a wireless access point communicating with the sensor nodes. The sink is represented by the black star in Figure 2.1 and the arrows in the figure shows a possible data path (route) from a sensor node in the network to the sink. The sensor network topology is frequently changed due to sensor nodes disappearing from the network or new sensor nodes being added into the network.

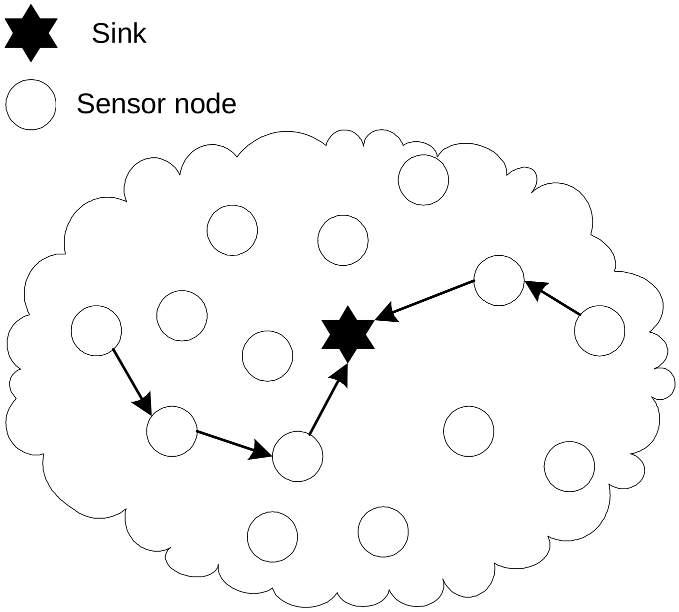

Figure 2.1: A typical sensor network topology with sensor nodes scattered over an area. The sensor nodes communicate and forward other sensor nodes' data to a sink placed somewhere in the network.

Having mobile sensor nodes also change the topology over time and the sensor network needs to handle these changes when they happen.

The typical sensor network topology in Figure 2.1 can be hierarchically divided and based on clusters, see Figure 2.2. One of the sensor nodes in a cluster becomes cluster head and the rest of the sensor nodes are called cluster nodes. The cluster nodes only communicate with the cluster head in their cluster. The sensor nodes not being cluster heads save energy because they only send their data to the cluster head, then they can turn off their radio in order to save energy, until they need to send again. A cluster node usually do not need to forward traffic from other sensor nodes. Being a cluster head is typically more energy consuming due to the increased communication between itself and its cluster nodes. The cluster head needs to listen for data from all its cluster nodes, and possibly also to listen for data from other cluster heads

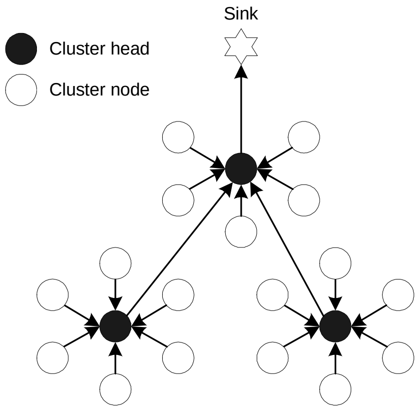

Figure 2.2: A typical hierarchical topology based on clusters. Sensor nodes communicate with its cluster head and the cluster head aggregates and/or fuses the received data before forwarding it to another cluster head or to the sink.

in the network. In order to distribute the extra workload of being cluster head, the task typically rotates among different sensor nodes in the network. In some sensor networks the sensor nodes' clocks might need to be synchronized. For example when a TDMA[2] scheme is used in order to handle data communication between the sensor nodes in the network. Two sensor nodes with exactly the same hardware will not have exactly the same clock frequency. The clocks typically deviate and cannot be synchronized perfectly and the clocks will over time drift apart. Therefore synchronization is needed between the sensor nodes in order to adjust the time in some applications.

2.4 Traffic reduction

In sensor networks using multihop, i.e., when data travels through several sensor nodes before reaching its final destination, the sensor nodes can process the data collected from its cluster nodes locally before forwarding it. In the next sections we describe two common ways to process data in order to reduce the data traffic and data size in the network.

---

[2] Time Division Multiple Access (TDMA) is described in Section 2.7.3.

2.4.1 Data aggregation

When multihop communication is used, e.g., as in the hierarchical topology described above, the sensor nodes can aggregate data collected from other nodes in order to reduce redundant data and thereby reduce the data traffic in the network [19]. If a cluster head and its cluster nodes have the same task in the network, e.g., measuring the surrounding temperature, they will probably send the same type of data to the sink. The data could, e.g., contain the sensed temperature, the cluster nodes' ID and the destination address of the data. Instead of sending several different data messages to the sink the cluster head aggregates the data into a single data packet. The resulting data packet could for instance contain all the sensor nodes' IDs and the sensed temperatures but only one of the same destination address is included in the data package. If several sensor nodes sense the same temperature they can be aggregated into one and further reducing the data packet.

2.4.2 Data fusion

If we continue the example from Section 2.4.1 but instead of reducing redundant information the cluster heads processes the data locally before sending the data packet further. The cluster head knows in advance that the user is interested in, e.g., the mean value of the surrounding temperature. Instead of sending a data packet with all the sensor nodes' IDs and all the measured temperatures, it computes the mean value of the temperature locally. The data being sent contains the mean value and the sensor nodes' IDs or possibly only an area ID. All the information needed to calculate the mean value of the temperature is the total sum, Σ, of all the measured temperatures divided by the number, n, of measurements. If there are several cluster heads between the sending cluster head and the user and they all are sensing values for a mean value calculation at the user, the cluster heads in between can continue to fuse data if the first cluster head includes the number n together with the mean value. This will reduce the size and the number of data packets.

2.5 Challenges in sensor networks

In this section we will discuss some challenges for sensor networks.
Some specific challenges are:

- **Wireless communication:** The sensor nodes communicate wirelessly with each other, e.g., with radio or with infrared/laser. The communication between nodes could be disturbed by external factors such as obstacles in line of sight when using infrared or, e.g., by other communicating nodes when using radio.

- **Limited energy supply:** Having wireless, adaptable unattended sensor nodes in harsh environments demands distributed algorithms to maintain the network. The sensor nodes have limited power supply and the sensor nodes need to conserve with the energy at their disposal. The most power-consuming activity is typically the communication between sensor nodes [26]. Hence, communication needs to be minimized in order to prolong the lifetime of the network as much as possible.

- **Prone to errors:** The sensor nodes are prone to errors [1, 33]. They could disappear due to fabrication errors, short-circuit caused by water leaks, lack of power or, e.g., animals/vehicles or humans breaking them.

- **Adaptable:** The sensor networks need to operate in very dynamic environments and with dynamic changes of the network. They are left unattended after deployment and thus the networks need to be very adaptable to the environment. The task a sensor node performs may change over time and the networks need to reconfigure themselves and be task adaptable, i.e., the sensor nodes might change the current task and perform another or get an additional task to the current one.

We address these challenges in papers A-C but sensor networks also have several other challenges not addressed in this thesis, for instance:

- **Security:** Some information from the sensor nodes should be protected. For example, the integrity of a patients' health in a hospital is important. Information of the health state from the sensor nodes on a patient should not be able to be read by unauthorized persons. Having an ad hoc collaborating sensor network in an area could for instance consist of several different sensor nodes from several different companies. Sensitive information might not be allowed to be forwarded by untrustworthy sensor nodes from other companies in the network.

  Data from sensor nodes triggering an intruder alarm for example should not get lost or disappear in the network. Some applications, such as intruder alarms or safety applications, need to be able to rely on that the information sent to the sink actually will be received.

- **Ad hoc:** Some sensor networks are situated in areas without infrastructure and after deployment, connect to each other in ad hoc manners. The sensor nodes could for example be thrown out from an airplane over the area to be monitored or the sensor nodes could be mobile and move around.
- **Identifier:** The sensor nodes are often densely deployed and in some application areas global identifiers are missing. Instead of identifiers the sensor nodes with a certain task/attribute answer a question from a user. The data itself instead of the actual sensor nodes' ID is of importance, data centric. A question to the sensor nodes in the network could be: Where are the nodes with temperature above $25^\circ\mathrm{C}$? In other areas the sensor nodes are divided into clusters where the cluster itself has an identifier but the individual sensor nodes have not.

The challenges described above introduce some energy tradeoffs to consider when optimizing the sensor network. A discussion on some of the energy tradeoffs are presented in Section 4.1.

2.6 Sensor node architecture and design

In order to meet the challenges and to be feasible to apply in the application areas described in Section 2.1 and Section 2.5, the sensor nodes need to be cheap, consume very little energy [16], and also be able to operate in densely deployed areas as well as being adaptable and flexible. In this section we describe the sensor node architecture and the design of the sensor network in more detail.

2.6.1 Sensor node architecture

A typical architecture of a sensor node can be divided into four units; processing unit, sensing unit, power unit and a radio unit that is able to both transmit and receive (transceiver), see Figure 2.3.

The onboard sensor unit consists of two subunits, sensors and analog-to-digital converter (ADC). The ADC converts analog signals from the sensors to digital signals used by the processing unit. There exists many different sensor types, such as [1]:

- seismic vibrations, low sampling rate magnetic, thermal, visual, infrared, acoustic and radar

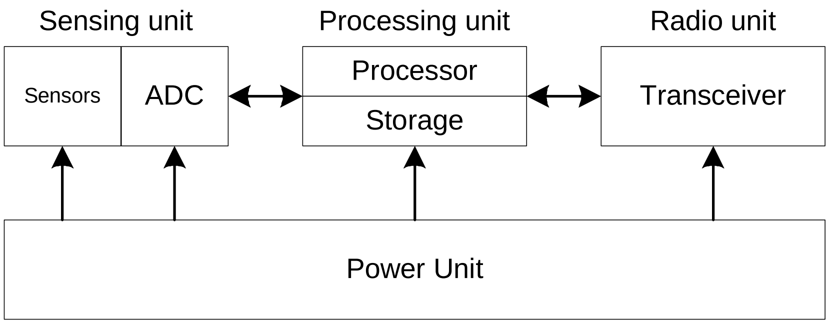

Figure 2.3: A typical architecture of a sensor node divided into four units.

monitoring a wide variety of ambient conditions like [11]:

- temperature, humidity, movement, light, pressure, soil make-up and noise.

The processing unit, usually a low speed CPU with small storage capabilities, performs tasks like routing, aggregation of sensed data etc. The transceiver unit communicates with the surrounding world and the power unit provides power to the other units. The power unit is typically a battery and an extra energy scavenging unit can be added to the battery, e.g., solar cells, prolonging the lifetime of the sensor node.

A mobility unit can in some cases be added to the sensor node as well as a localization unit, e.g., a global positioning system (GPS). All these units might need to fit into a combined unit small as a matchbox [9] or even within a cubic millimeter [35]. It is of great importance that all units together consumes a extremely small amount of energy in order to prolong the lifetime of the network. The sensor nodes should be able to operate unattended and be able to adapt to the current environment.

The desired cost of a sensor node should be less than one US dollar [27]. The sensor nodes need to be cheap in order to make unattended, densely deployed sensor nodes cost-justified compared to traditional sensors.

2.6.2 The design of the sensor network

The sensor network design, as demonstrated, should reduce the installation cost and the network should be fault tolerant, scalable, flexible and self-organizing.

As mentioned in Section 2.5, the sensor nodes are prone to errors and in most applications it is important that the network is fault tolerant and does not get affected by failing sensor nodes in the network [15]. Some of the sensor nodes in the network could be mobile or be moved by hand or by other external factors. Data should be rerouted through other sensor nodes if existing routes fail and adjacent sensor nodes could, e.g., take over the failing node's task.

A typical network could span from hundreds up to thousands of sensor nodes in the network [30]. The network density can scale up to 20 sensor $\mathrm{nodes/m^2}$. In industrial applications where the sensor nodes, in for instance monitor machinery, there can be 300 sensor nodes within 5 x 5 $\mathrm{m^2}$ and 3000 in a 100 x $100\mathrm{m^2}$ [30]. The density is application specific, thus we need scalable schemes to handle the dynamics in the sensor network. It is also important that the sensor networks are autonomous in handling the dynamics and reconfigure themselves when needed.

Some time after deployment of a sensor network in an area, new nodes need to be added to replace failing nodes. Flexible schemes handling new sensor nodes and re-arranging the network to the new conditions are necessary. There can be frequent changes in the network; sensor nodes can be jammed in some way or can not be reached, out of range, they malfunction or have no energy left. The task a sensor node should perform can change over time depending on the application. We need a self-organized network handling the changes in the network.

In some application areas, it is not important that the sensor nodes send their messages as in traditional address-centric networks, with ID and sensed value. In some applications, the areas where a certain phenomenon occurs, e.g., the areas where the temperature exceed $25^\circ\mathrm{C}$, are interesting. The sink broadcasts the query and the sensor nodes with a temperature over $25^\circ\mathrm{C}$ send back a message to the sink. This type of communication can be either broadcasting-based or attribute-based. If it is attribute-based, the sensor nodes need to be divided into different attributes that are used as an identifier, attributes such as temperature, humidity, pressure etc. The sensor nodes could be divided into a geographic areas where the area is of importance/interest and not the sensor nodes themselves. If several sensors run the same task, e.g., monitor the same rolls on a machine, and any-casting is used they can all be associated with the same identifier. Not all of the sensor nodes need to be awake at the same time when using any-casting. Some of the sensor nodes can stay asleep until they are needed and only one of the sensor nodes associated with the identifier needs to answer the query.

2.7 Medium access mechanisms

In this section we briefly describe some important medium access mechanisms mentioned in the thesis. This section addresses readers not familiar with data communication protocols. The purpose of the medium access mechanism is to, e.g., avoid, reduce or handle communication collisions in the network.

2.7.1 Carrier Sense Multiple Access

The Carrier Sense Multiple Access (CSMA) protocol tries to detect the absence of other traffic in the network before transmitting its own [32]. Using CSMA will not entirely prevent collisions. Different techniques exist to handle possible collisions and we will briefly describe two of them.

If two nodes try to send at nearly the same time using pure CSMA, none of them will detect the other's carrier and a collision occurs. The nodes may detect that a collision has occurred if acknowledgements (ACKs) are used and the sender gets a timeout due to the absence of an ACK from the receiver. However, the nodes that are causing the collision continue to send their entire data packet, thus, wasting bandwidth.

CSMA with Collision Avoidance (CSMA/CA) uses a different strategy [8]. After a node ready for transmission has sensed the absence of other traffic, the node informs the receiver that it is intending to send. The receiving node replies back to the node if no other traffic is sensed. If traffic is sensed the transmitting node waits a random deferral time before informing again. The actual message is sent after getting a reply from the receiver. Collisions may occur with CSMA/CA when two nodes ask to send to two different receivers and the two receivers are not able to hear the other transmitting node's message in time before both of them replies back. The delay time is increased when using CSMA/CA.

In CSMA with Collision Detection (CSMA/CD), the nodes sense for carrier and start to send if there is no traffic. The nodes are able to detect when a collision occurs. When a collision occurs, the node sensing the collision stops to transmit its data immediately and sends a jamming signal instead. The node waits a random deferral time before trying again. Using CSMA/CD in wireless networks is difficult since not all the nodes in the network can be assumed to hear all the others. Thus, e.g., the jamming signal may not be correctly received by all nodes.

2.7.2 Frequency Division Multiple Access

In Frequency Division Multiple Access (FDMA), the frequency spectrum is divided into several channels. The nodes use different channels to communicate without interference from other communicating nodes. Take the FM radio for instance, different radio stations send on different frequencies without disturbing each other. The user adjust the radio receiver to the station's frequency he/she wishes to listen to. When frequency is divided, the bandwidth for each channel is reduced, hence, the data throughput is decreased.

2.7.3 Time Division Multiple Access

Instead of dividing the frequency, the time can be divided into cycles and slots as in Time Division Multiple Access (TDMA). A cycle consists of several slots and is repeated over and over again. Each node gets its own slot, i.e., a time frame, where it is allowed to send its data. The node starts at the beginning of its slot and need to finish before the slot ends. A node can get one or several slots each cycle. Using the radio example in the FDMA section, a typical radio broadcasting station can be seen as music songs played after each other mixed with commercials. The songs and commercials get the whole frequency band and do not disturb each other.

2.7.4 Code Division Multiple Access

In Code Division Multiple Access (CDMA), all the sensor nodes can use the whole frequency spectrum at all time [32]. Multiple data transmissions can simultaneously be transferred and the data transmissions are separated using coding theory. Each data bit time is subdivided into intervals called chips, typical 64 or 128 chips per bit. All sensor nodes get a unique code or chip sequence and they use this code to encode the transferred data. To send a 1 it send its code sequence and when sending a 0 it sends the one's complement of the code sequence. The sensor node receiving the message decode the message using the sending sensor nodes' code. The sensor nodes need to know all the other sensor nodes' codes in order to decode the transmitted data.

Chapter 3

Existing infrastructure and problem formulation

In this chapter we briefly explain what an existing infrastructure is and introduce the research area of this thesis.

3.1 Existing infrastructure

We define an existing infrastructure as a computer network together with its computers and peripheral equipments. Such existing infrastructures can be found in, e.g., hospitals and industrial buildings. The existing infrastructure have computers, laptops, Personal Digital Assistants (PDAs) and phones connected to a high-speed Local Area Network (LAN), see Figure 3.1.

Some of the computers in the LAN are regular Personal Computers (PCs). The PCs are often stationary computers, i.e., they are, e.g., workstations on a desk and static connected to the LAN by wire. Whereas laptops and PDAs often are mobile and can be either connected to the LAN by wire or wirelessly through a wireless access point. Phones using the network as carrier for conversations can be either phones on the desk or wireless mobile phones using the wireless access points. In order to organize the network, computers are needed within the LAN. Some of the computers handle traffic flows and make sure that the data traffic reach its destination in the network. Other computers acts as file servers and their task are to store data, accessible for many users within the LAN. The data could be, e.g., patient journals, electrical schemat-

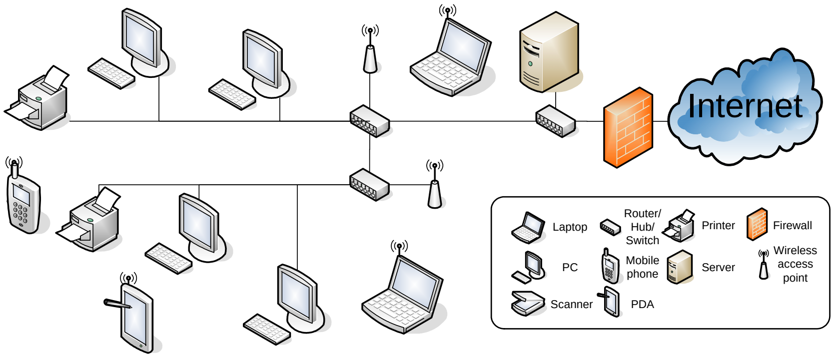

Figure 3.1: An example of how a local area network in hospitals and industrial buildings could look like.

ics, source code or other shared documents. Some servers handle requests for email, print-outs and other network services from the users within the LAN. Some servers handle the accessibility to/from the Internet. The servers could if desired, stop the access to the LAN from user outside on the Internet or other LANs with a firewall. But they could still allow users within the LAN to access the Internet.

3.2 Background

Comparing the computers in existing infrastructure to the sensor nodes in sensor networks, the computers do not have scarce resources as the sensor nodes. The computers have high-speed CPUs capable of running computation-intensive applications. They have plenty of memory where they, e.g., can store information from computations, data and process states. Their power supply is "unlimited"[1] as they are connected to the mains and the network is typical a high-speed wired network designed for heavy traffic flows.

As the employees more and more get equipped with laptops and other peripherals with wireless technology, we will certainly see an increased usage of wireless communications in hospitals and industrial buildings. The employ-

---

[1] For simplicity we will use the word unlimited throughout the thesis when talking about the power supplies of the existing infrastructures' computers.

ees do not need a fixed work desk with a stationary computer. The employees could for instance bring their computers from the work desk in their office to the meeting rooms and down to the laboratory, wirelessly connected to the companies' LAN. It is our belief that the number of wireless access points will increase even more in the industrial and hospital environments as more and more electronic devices with wireless techniques are developed.

The industry forecasts an explosive growth in the use of sensor network applications in industry in the near future [3]. The sensor nodes in industry applications are intended to replace traditional wired sensors. New applications, and applications considered too expensive before or even not possible will now become possible. The sensor nodes will enlarge the application areas in the industry, adding low-cost and mobile sensor nodes in areas not cost-justified earlier.

3.3 Problem formulation

In this thesis we investigate if existing infrastructure, such as described above in Section 3.1, can aid in organizing a wireless sensor network scattered over a large area.

For example, consider two disjoint sensor networks performing the same task and delivering information to a sink within its own network. Remember that we in Section 2.3 said that, a sink in sensor networks usually consists of a high-performance computer with a wireless access point. Instead of sending the information to a sink each, the sensor nodes could send the sensed information through a close by access point in the LAN to one single shared sink. Some of the computers within the existing infrastructure could act as base stations and help the sensor networks to organize themselves and, e.g., manage clock synchronization, routes and schedules. The base stations could handle tasks considered too energy consuming to perform for the sensor nodes themselves.

A lot of the communication between sensor nodes in a purely sensor network is communication to maintain routes and topology changes. By centralizing the maintenance of data routes and topology changes/optimizations in a sensor network, we can prolong the lifetime of the sensor network. If the existing infrastructure has knowledge about the whole network and has unlimited energy, it can perform optimizations not energy cost-effective in a purely sensor network by centralizing distributed algorithms like routing and topology changes.

Exposed sensor nodes forwarding data, can drain their batteries. Therefore, the sensor network needs to have routing algorithms adjusting the routes in order to distribute the workload. But to distribute the workload between the sensor nodes in the network, we need intelligent distributed algorithms and network information exchange between the sensor nodes in order to maintain the routes. Hierarchical topologies like clusters with cluster heads and cluster nodes might need to be rearranged depending on, e.g., the number of sensor nodes and the amount of energy in a cluster. In order to make these topology changes the sensor nodes need to communicate with each other. Our hypothesis is:

Centralizing communication-intensive algorithms like routing, cluster formation and sensor network optimizations will save energy and thus prolong the lifetime of the sensor network.

Chapter 4

The architecture of AROS

AROS, Asymmetric communication and ROuting in Sensor networks, see Figure 4.1, is an architecture based on the problem formulation in Section 3.3. The architecture uses existing infrastructure as support for wireless sensor networks. The infrastructure is mostly static, but there can exist mobile base stations. The infrastructure is used to build base stations that help the sensor nodes with energy consuming tasks. The infrastructure can consist of regular computers, PDAs, cellular phones or small embedded systems. The base stations are connected to each other by wire, wirelessly or both, creating a backbone for the sensor nodes. The base stations have unlimited energy and long range wireless communication capacity. Having unlimited energy, the base stations can always keep their radio on and listen for incoming data from the sensor nodes in their network. The base stations have high speed processors and plenty of memory, in comparison to the sensor nodes.

The sensor nodes in the network have scarce resources and the communication range is therefore more limited than that of the base stations, because of the limited energy availability. In order to communicate with the base station some of the sensor nodes might need other sensor nodes to forward their data. The sensor nodes could be mobile and move themselves or be moved by hand, if the task changes over time. Some of the sensor nodes may run out of energy or could be moved out of radio range from the base station when moved. Other sensor nodes could malfunction due to fabrication errors, short-circuits etc.

The sensor network needs to be robust, dynamic and flexible. AROS sensor nodes' use multihop forwarding in order to communicate with the base station and possibly, also with other sensor nodes in the network. By allowing

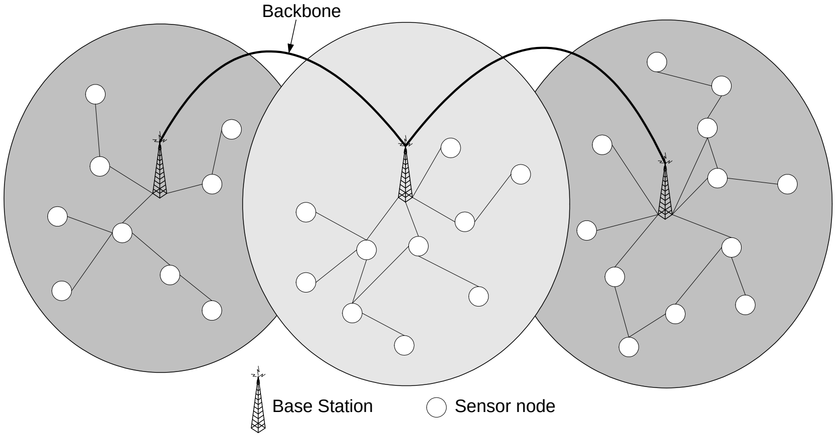

Figure 4.1: The AROS topology with three base stations and scattered sensor nodes. The base stations are connected with each other in a backbone. Possible data paths from the sensor nodes in the cluster towards the base station are shown as an example.

multihop communication the area covered by one base station can be determined by the range of that base station's radio, rather than by the range of the sensor nodes' radio. In order to reduce data communication in the network, an hierarchical layered topology is used with cluster heads and cluster nodes. The cluster heads should, if possible, aggregate or/and fuse data in order to minimize data traffic in the network.

4.1 Application tradeoffs

With the challenges described in Section 2.5 some energy tradeoffs need to be considered when optimizing/designing a sensor network. For instance, if the sensor network application is an application with high QoS demands, e.g., an intruder alarm, minimizing the delay-time for the data (alarm) from the sensor nodes to the sink might be the most important task for the sensor network. The sensor nodes possibly consume more energy because they need to listen for traffic to forward or send data longer distances in order to reach the sink. High QoS in delivering important data to the sink might include that the sensor nodes need to send an ACK to the sending sensor node after receiving a message, at the energy consumptions' expense. Applications optimizing the sensor network for longevity might need to tolerate longer delay-time for the data to reach its destination. Mobile nodes in the network will increase the communication between the sensor nodes in order to maintain the network, hence consuming more energy. Safety issues also introduce energy tradeoffs. Encryption of data increases the workload of the CPUs at the sensor nodes and possibly increases the data size. Reliable data paths from a sensor node to the sink increase the communication needed in order to establish guarantees. The sensor node density could be of importance when sensor nodes forward data, in dense sensor networks, sensor nodes does not need to send their data long distances, hence, saving energy.

These are some examples of energy tradeoffs needed to be considered when optimizing the sensor network. Minimizing the delay-time could increase the energy consumption and minimizing the energy consumption could increase the delay-time. The energy tradeoffs need to be considered carefully depending on the application.

4.2 The AROS vision

Our vision is, by using existing infrastructure, the lifetime of the sensor networks will be prolonged. By using existing infrastructure the communication exchange between the sensor nodes can be reduced and hence, the sensor nodes can save energy. Mobile sensor nodes changing cluster or changing from one base station to another, will be handled by the base stations instead of by distributed algorithms performed by the sensor nodes themselves. The base station will, depending on the application, calculate the best energy tradeoffs for the sensor network, as described in Section 4.1.

The base stations in AROS have long radio coverage and instead of sensor nodes running complex distributed multihop clock synchronization algorithms, the clock synchronization can be handled by the base stations. The base stations can handle routing issues for the sensor nodes and by, i.e., monitoring the sensor nodes' energy level, the base station can change routes from sensor nodes with low energy levels to sensor nodes with higher energy levels. This will save sensor nodes from draining their energy when being highly exposed to forward data from other sensor nodes, and it will avoid data losses from vanishing routes. Topology changes and topology optimizations are handled by base stations with high-speed processors and plenty of memory. If the base station knows the energy level, the position and task of all the sensor nodes in the network, it can perform topology optimizations not energy cost-effective if performed by the sensor nodes themselves. The base stations can communicate directly with the sensor nodes in the network. A query from a base station to a specific sensor node or region for instance, can be asked directly to the sensor node or region without involving other sensor nodes in between. The sensor nodes just need to focus on their assigned tasks until the base station inform them about task changes or/and topology changes they need to know about. If the sensor nodes know their tasks and when to communicate with each other, they can turn off their radio in between and thereby save energy.

In order for the sensor nodes to be able to turn off the radio in a forwarding sensor network without dropping data, knowledge about the communication between the sensor nodes need to be known in advance. One possible solution to enable predefined communication between sensor nodes is to use the TDMA protocol. The base stations can calculate a schedule for the sensor nodes and supply the sensor nodes with the information they need to know. To schedule a cluster-based sensor network with pure TDMA can increase the delay-time for data to be received at the base station from the outermost sensor nodes. The scheduler can, e.g., divide the spectrum into different channels like in FDMA, and dedicate a separate channel for each cluster. This makes it possible for several clusters to communicate in parallel and thus minimizing the length of the TDMA schedule. A combination of TDMA and FDMA could be a solution to some scheduling problems.

The base station could handle sensor networks with different types of application demands. Sensor nodes with low demands on the delay time can be mixed with sensor nodes with high demands on the delay time or they could be divided into completely different sub-networks. For example, the base station can divide the different sensor nodes into different sub-networks and build separate routes and schedules for different applications if necessary. Or, it can use cluster heads from one application with low interest of saving energy to forward data from sensor nodes with high interest of saving energy. Depending on the application, the base station can calculate an optimal schedule with optimal routes and sub-networks for the sensor nodes.

As we mentioned earlier, the base stations have long distance communication capabilities and are capable of transmitting data directly to all sensor nodes in their network. The sensor nodes on the other hand might not be able to communicate directly with its base station but need other sensor nodes to forward their data. We call this asymmetric communication and when all the sensor nodes in the network can communicate with the base station directly we call that communication symmetric.

Chapter 5

Related Work

In this section we discuss some related work and how they relate to the AROS architecture.

5.1 The LEACH project

LEACH (Low-Energy Adaptive Clustering Hierarchy) [14] and AROS are compared throughout the papers in this thesis. We have chosen to compare AROS to LEACH for a number of reasons. LEACH is a well known TDMA cluster-based sensor network architecture and the architecture is fairly simple to compare some of the most important aspects like energy usage per message and network lifetime. LEACH sends data frequently to the sink or base station without complex algorithms such as thresholds values, see Section 5.2. AROS can do all the things that LEACH can do and more. AROS can handle safety issues, routing of data, mobility, handover of sensor nodes from one base station to another, several different types of sensor networks, clock synchronization, reorganization of the sensor network and sensor network optimizations. In this thesis we have restrained AROS functionality to that of LEACH in order to be able to compare the architectures.

LEACH 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

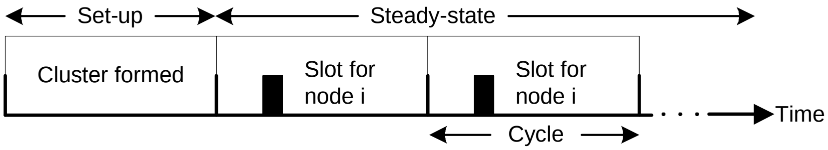

Figure 5.1: The TDMA scheme in the LEACH architecture.

head needs to receive data from all cluster members in its cluster and then send the data to the base station. All sensor nodes in the network have the potential to become cluster head at some point in time. A TDMA round consists of one set-up phase and a steady-state phase. The TDMA scheme starts every round with a set-up phase to organize the clusters, see figure 5.1. After the set-up phase, the system is in a steady-state phase for a certain amount of time. The steady-state phase consist of several cycles where all sensor nodes have their slots periodically. The sensor 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.

5.1.1 LEACH-C

LEACH-C (LEACH-Centralized) [13] 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 phase as LEACH. During the set-up phase, the base station receives information from each sensor node about their current location and energy level. According to [13], the sensor 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 [23] to search for near-optimal clusters. Once the clusters are created, the base station broadcasts the information to all the sensor nodes in the network. After receiving the message the sensor node goes to sleep until it is time to transmit data to its cluster head.

5.1.2 LEACH-F

A further development is LEACH-F (LEACH with Fixed clusters) [13], which is based on clusters that are formed once in the first setup-phase - and then the clusters remain fixed. After clusters have been established, the cluster head position rotates among the sensor 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, no extra communication from the base station is needed. 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 sensor node mobility.

5.2 TEEN and APTEEN

TEEN, Threshold sensitive Energy Efficient sensor Network protocol [21] is an extension of the LEACH project, described in Section 5.1. TEEN is a time critical protocol best suited for time critical applications. The protocol is, as LEACH-C, cluster based with base stations maintaining the cluster head grouping. The base station can communicate directly with the sensor nodes but the sensor nodes are, as in AROS, not always able to communicate directly with the base station.

The TEEN protocol introduces a hard and a soft threshold. The hard threshold is a threshold value a sensor node needs to sense before sending data to the cluster head. For instance, sensor nodes sensing temperature need to read $25°\mathrm{C}$ before sending data to the cluster head. The soft threshold is the maximum difference the value is allowed to differ from the hard threshold before sending data to the cluster head. For instance, once sensing $25°\mathrm{C}$ the temperature must differ $\pm 2°\mathrm{C}$ before sending a new data to the cluster head. The sensor node do not send any data until sensing the hard threshold value leaving the base station ignorant of the sensor nodes' state. The base station can not assume that the sensor nodes are alive and working if it does not get data from the sensor nodes. Even though the sensor node is not sending data to the cluster head, it senses its surroundings and possibly it even performs some sort of computations using energy.

TEEN differs from AROS in several ways. Once the hard threshold has been reached the sensor node turn on its radio and sends the data to the base station. Collisions might occur between concurrent communicating sensor nodes.

The authors say that to avoid collisions between concurrent communicating sensor nodes, TDMA scheduling or CDMA could be used. Using TDMA will increase the delay time for the data to be received at the base station. The lifetime of the sensor network in TEEN depends on how often the cluster nodes communicate with the base station. The number of collisions in the network increases as the communication increases. Intensive communication can even increase the amount of energy used per bit, compared to the LEACH architecture.

The cluster heads in the network send out data with hard and soft threshold attributes to the sensor nodes. In AROS the base station handles all communication to the sensor nodes in the network. Further, the cluster heads in TEEN can not turn off their radio as AROS' cluster heads can. The cluster head in TEEN can receive data from its cluster nodes at any time. In AROS the cluster heads know when the cluster nodes send their data, and thus, the cluster heads can turn off their radio when not scheduled to receive or transmit data.

APTEEN, Adaptive Periodic Threshold sensitive Energy Efficient sensor Network protocol [22], uses the same architecture described above and the cluster nodes can be scheduled to send data periodically to the base station. As in TEEN, the cluster heads distribute the schedule to its cluster nodes. APTEEN uses TDMA to schedule the cluster nodes, eliminating the radio collisions between cluster nodes. Thus, increasing the delay time in the network for time-critical data. The clusters use different CDMA codes in each cluster an a commonly used code to talk to other cluster heads and the base station. The cluster heads can not turn off their radio in APTEEN since another cluster heads can transmit data any given time.

Comparing TEEN and APTEEN to AROS could be useful if the AROS architecture only suppose to handle realtime applications. However, AROS can handle several types of applications. Therefore, we choose to compare AROS to LEACH since we are interested in minimizing the energy used per bit in the network.

Chapter 6

Summary of the papers and their contribution

In this thesis we have studied some of the research questions from Section 3.3. The contributions in the thesis have been published at peer-reviewed international conferences. The vision described in Paper A has been presented at three different conferences, Euromicro-04, SNCNW-04 and Medicinteknikdagarna-05 in the poster session. Paper C is a technical report of the paper to be presented in Med-Hoc-Net 2006 [25]. In addition to the scheduler presented in [25], the technical report also includes a simulation evaluation of the TDMA-scheduler. The papers included in this thesis have only been modified to suit the typography of this thesis and discovered typos have been corrected.

We believe that existing infrastructure can aid in organizing sensor networks. In order to justify our believes we presented our vision to the community in order to solicit information about the proposed architecture. After discussing the vision at several conferences we performed a static simulation study comparing our architecture to another architecture, LEACH. We needed to constrain AROS to their simulation setup in order to make a comparable comparison. In order to dynamically construct clusters we presented a TDMA scheduler and once again compared AROS to LEACH in order to strengthen our believes that existing infrastructure can aid in organizing sensor networks. Once again AROS was simulated with constraints in order to make a comparable comparison.

In the papers B and C we say that AROS is built on the LEACH architecture. We extend LEACH by introducing asymmetric communication into the LEACH architecture. But AROS is capable of much more than what the LEACH architecture is.

6.1 Paper A: Using Existing Infrastructure as Proxy Support for Sensor Networks

Using Existing Infrastructure as Proxy Support for Sensor Networks, Jonas Neander, Mikael Nolin, Mats Björkman, In 16th EUROMICRO Conference on Real-Time Systems (ECRTS 04), Work in progress, Catania, Italy, June, 2004

In this paper we propose a semi-centralized sensor network approach where existing, powerful, infrastructure can be used to off-load sensors and prolong network lifetime. The semi-centralized sensor network AROS uses base stations in order to off-load the sensor nodes. We describe the vision of the AROS architecture and discuss problem areas and possible solutions.

The contribution of this paper is the vision of the AROS network architecture.

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

6.2 Paper B: Asymmetric Multihop Communication in Large Sensor Networks

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

In this paper we provide an initial simulation study comparing asymmetric multihop communications and symmetric single-hop communications. The asymmetric multihop communication is represented by AROS and the symmetric single-hop is 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 long distance networks, forwarding data is more energy efficient than sending it directly to the base station.

The contribution of this paper is to 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.

I was the main driving author of this paper and I wrote most of the text for the paper. My co-worker and co-writer Ewa Hansen and I have implemented AROS and performed the simulations in NS-2.

6.3 Paper C: A TDMA scheduler for the AROS architecture

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

Paper C is a technical report of the paper to be presented in Med-Hoc-Net 2006 [25]. In this paper we present a TDMA scheduler for the AROS architecture enabling dynamic network configurations. We continue the simulations performed in paper B with the comparison between asymmetric multihop communication and symmetric single-hop communication but this time with dynamic network configuration. We also provide a simulation comparison between the static network configuration in paper B and the new dynamic network configuration.

We show that asymmetric multihop communications with the TDMA scheduler prolongs the lifetime of the sensor nodes with dynamic network configurations in long distance networks. In our simulations we have studied how dynamic network clustering in AROS, with non-mobile nodes, affects the amount of data received by the base station. We also show that AROS is better than LEACH in collecting data to a base station with the same total amount of energy for long distance networks. We also show that AROS performs as well or better than LEACH-C in small networks.

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

Chapter 7

Conclusions and future work

In this thesis we have presented an architecture called AROS that uses existing infrastructure to aid a sensor network with scarce resources. The existing infrastructure can be situated in, e.g., hospitals and industrial buildings. The existing infrastructure can aid in prolonging the lifetime of the sensor network, as the existing infrastructure has unlimited energy, long range radio capacity and high-speed computers. Prolonging the lifetime is achieved by centralizing some of the energy consuming tasks that before were performed by the sensor nodes themselves.

Not all sensor nodes are assumed to be able to communicate directly with the infrastructure in AROS, some sensor nodes need other sensor nodes in order to forward its data. Forwarding data from other sensor nodes in sensor networks is usually one of the most common forms of collaboration between sensor nodes. Experimental measurements indicate that communication cost in wireless ad-hoc networks is at least two orders of magnitude higher than computation costs in terms of consumed power and according to [26], the most energy-consuming activity in sensor networks.

We have shown with initial simulations that the AROS architecture is suitable for prolonging the lifetime of sensor nodes in the sensor network. The simulations are compared to a well known symmetric cluster based architecture called LEACH. Comparing AROS to LEACH, forced us to restrain AROS to that of LEACH in order to do a fair comparison. The comparison shows that AROS with static configuration performs at least as well as LEACH in small networks, less than 100x100m, and up to 97 % better in large networks, 400x400m. We have shown that AROS still has got 88 % of its sensor nodes alive when all the sensor nodes in LEACHs' network have used all of their energy and demised.

In our simulations, we have also studied how dynamic network clustering in AROS using a TDMA scheduler and non-mobile sensor nodes, affects the amount of data received by the base station. We have shown that AROS is better than LEACH-C in collecting data to a base station with the same total amount of energy for long distance networks. We have also shown, that AROS performs as well or better than LEACH-C in small networks.

Future work

In this thesis we have shown simulations from statically and dynamically configured networks. The round time and the number of cluster heads have been fixed in order to be able to compare AROS to LEACH. In future work we will evaluate what types of scenarios AROS is suitable for by increasing the simulation domain. We are planning to perform thorough simulations of AROS where we lift some of the restrictions placed on AROS in order to compare it against LEACH. Two such important restriction is the number of cluster heads and the round time. Our belief is that AROS can perform even better when being able to change the number of cluster heads and being able to vary round times. Also, initial results show that network lifetime can be improved when distributing the cluster heads more evenly over the network [12]. We will further investigate methods to distribute remaining energy evenly over the whole sensor network, in order to maximize the lifetime. We will implement intelligent routing algorithms, better clustering formations and investigate how AROS behaves with different number of sensor nodes in the network. Furthermore, we will investigate other parameters than the number of packets received at the BS. An example result metric include how network lifetime is correlated to the delay time in the network.

We are planning to do a simulation comparison between AROS and TEEN and APTEEN in order to evaluate AROS further and see how AROS performs in comparison to other architectures than LEACH. We will also implement the AROS architecture in a real network and test how AROS behaves in a real sensor network.

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] Automated Local Evaluation in Real Time (ALERT). World Wide Web, `http://www.alertsystems.org`, April 2006.

[3] N. Baker. ZigBee and Bluetooth strengths and weaknesses for industrial applications. *Computing & Control Engineering Journal*, 16(2):20–25, April/May 2005.

[4] A. Baptista, T. Leen, Y. Zhang, A. Chawla, D. Maier, W. Feng, W. Feng, J. Walpole, C. Silva, and J. Freire. Environmental Observation and Forecasting Systems: Vision, Challenges and Successes of a Prototype. *International Society for Environmental Information Sciences Annual Conference (ISEIS'2003)*, July 2003.

[5] A. Bharathidasan and V.A.S. Ponduru. Sensor Networks: An Overview. Technical report, Dept. of Computer Science, University of California at Davis, 2002.

[6] C. Chong and S.P. Kumar. Sensor Networks: Evolution, Opportunities, and Challenges. *Proc IEEE*, August 2003.

[7] CORIE. World Wide Web, `http://www.ccalmr.ogi.edu`, April 2006.

[8] G. Coulouris, J. Dollimore, and T. Kindberg. *Distributed Systems: Concepts and Design*. Addison-Wesley, third edition, 2001. ISBN 0201-61918-0.

[9] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the Physical World with Pervasive Networks. *Pervasive Computing*, (1):59–69, 2002.

[10] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the World with Wireless Sensor Networks. Salt Lake City, Utah, May 2001. International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001).

[11] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next Centuru Challenges: Scalable Coordination in Sensor Networks. *ACM MobiCOM '99*, 1999.

[12] E. Hansen, J. Neander, M. Nolin, and M. Björkman. Energy-Efficient Cluster Formation for Large Sensor Networks using a Minimum Separation Distance. In *The Fifth Annual Mediterranean Ad Hoc Networking Workshop*, Lipari, Italy, June 2006.

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

[14] 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).

[15] G. Hoblos, M. Staroswiecki, and A. Aïtouche. Optimal Design of Fault Tolerant Sensor Networks. In *Proceedings of the 2000 IEEE International Conference on Control Applications*, pages 25–27, Anchorage, Alaska, USA, September 2000.

[16] J. M. Kahn, R. H. Katz, and K. Pister. Next Century Challenges: Mobile Networking for "Smart Dust". *ACM MobiCom '99*, pages 271–278, 1999.

[17] J. M. Kahn, R. H. Katz, and K. Pister. Emerging Challenges: Mobile Networking for "Smart Dust". *Communication and Networks*, pages 188–196, September 2000.

[18] H. Karl and A. Willig. A short survey of wireless sensor networks. Technical Report TKN-03-018,, Telecommunication Networks Group, Technical University Berlin, Oct 2003.

[19] B. Krishnamachari, D. Estrin, and S. B. Wicker. The Impact of Data Aggregation in Wireless Sensor Networks. In *Proceedings of the 22nd International Conference on Distributed Computing Systems, ICDCSW*, pages 575–578, Washington, DC, USA, 2002. IEEE Computer Society.

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

[21] 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.

[22] 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.

[23] 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.

[24] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava. Coverage Problems in Wireless Ad-hoc Sensor Networks. *INFOCOM 2001, IEEE Computer and Communications Societies*, 3:1380–1387, April 2001.

[25] J. Neander, E. Hansen, J. Mäki-Turja, M. Nolin, and M. Björkman. Prolonging network lifetime in long distance sensor networks using a tdma scheduler. Lipari, Sicily, Italia, Jun 2006. The Fifth Annual Mediterranean Ad Hoc Networking Workshop. Med-Hoc-Net 2006.

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

[27] J. M. Rabaey, M. Josie Ammer, J. L.da Silva Jr., D. Patel, and S. Roundy. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer*, pages 42–48, July 2000.

[28] P. Rentala, R. Musunuri, S. Gandham, and U. Saxena. Survey on sensor networks. Technical Report UTDCS-33-02, University of Texas at Dallas, Dallas, Texas, 2002.

[29] L. Schwiebert, S. Gupta, and J. Weinmann. Research Challenges in Wireless Networks of Biomedical Sensors. *ACM Mobicomm 2001*, pages 151–165, 2001.

[30] E. Shih, S.H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks. *ACM SIGMOBILE*, July 2001.

[31] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. Protocols for Self-Organization of a Wireless Sensor Network. *IEE Personal Communications*, Oktober 2000.

[32] A. S. Tanenbaum. *Computer Networks*. Prentice-Hall, Inc, Upper Saddle River, New Jersey 07458, third edition, 1996. ISBN 0-13-394248-1.

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

[34] M. A. V. Vieira, C. N. Coelho Jr., D. C. da Silva Jr., and J. M. da Mata. Survey on Wireless Sensor Network Devices. *ETFA '03*, pages 537–544, September 2003.

[35] B. Warneke, M. Last, B. Liebowitz, and K.S.J. Pister. Smart Dust - Communicating with a Cubic-Millimeter Computer. *IEEE Computer*, 34:44–51, Jan 2001.

II

Included Papers

Chapter 8

Paper A: Using Existing Infrastructure as Proxy Support for Sensor Networks

Jonas Neander, Mikael Nolin and Mats Björkman
In 16th EUROMICRO Conference on Real-Time Systems (ECRTS 04), WiP,
Catania, Italy, June, 2004

Comments to the paper

In this paper the base stations were called proxies and sinks. We thought of several types of nodes in the network, where some nodes were more powerful than others. The more powerful nodes could offload the smaller nodes, thus called proxy. The data were sent to a sink somewhere in the network. Today the proxy and sink are the same and when we compared AROS to LEACH in Paper B, we change it to base station in order not to confuse the readers.

Abstract

In many environments where communications infrastructure already exists, e.g., factories and hospitals, sensor networks applications are becoming increasingly interesting. We present our ongoing work aimed at developing a network architecture using such existing infrastructure as proxy support for sensor networks. The proposed topology is asymmetric in its communication, i.e., the proxy can reach its sensor nodes in one hop but there is no guarantee that the sensor nodes reach the proxy directly. Thus we propose to divide the topology hierarchically.

Too handle sensor nodes with different demands and to save energy at the sensor nodes we propose to schedule the sensor nodes with Time-Division Multiple Access (TDMA). We outline a proposed network architecture and point out important research issues that must be addressed. One of the main purpose of this paper is to solicit feedback on our proposed network architecture.

8.1 Introduction

In this paper we present work in progress in the area of architecture design for sensor networks. With the growing interest in sensor networks, efficient communication infrastructures for such networks are becoming increasingly important. A sensor node is typically a tiny computer with limited computation resources and limited power supply, using on-board sensors to sense the surrounding environment, and using a wireless communication system to report to a network connection point (a network *sink*) [1, 2, 3].

Sensor networks are designed for many purposes. Among the interesting application areas are environmental surveillance and surveillance of equipment or persons in e.g. factories and hospitals. Common for all application areas are that sensor nodes are left unattended after deployment, that communication is wireless and the power supply is limited.

Having unattended sensor nodes with limited power supplies implies that one important feature of sensor networks is robust functionality in the face of network nodes dropping out of the network after some time of activity. 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. The network topology is thus dynamic even if the sensor nodes not necessarily are mobile.

Some sensor nodes will not be able to directly communicate with the network sink. The traffic from these sensor nodes must be forwarded by other sensor nodes, hence routing schemes are necessary. Routing of traffic through other sensor nodes will however increase the power consumption of the forwarding sensor nodes. Therefore, routing decisions must be carefully evaluated in order to maximize network lifetime.

The main research focus in sensor networks has been on building networks consisting of sensor nodes only. These, peer-to-peer networks rely on energy draining and complex distributed algorithms to establish e.g. network topology and membership. In this paper, however, we are proposing a semi-centralized approach where existing, powerful, infrastructure can be used to off-load sensors and prolong network lifetime.

The outline of this paper is as follows: In section 8.1.1 we motivate the use of a proxy backbone in our architecture and in section 8.1.2 we present related work done with TDMA in sensor networks. In section 8.2 we list some important problem issues in sensor networks and in section 8.3 we propose our asymmetric topology proxy backbone architecture for sensor nodes.

8.1.1 Proxy Solution

In order to lower the risk of a sensor node draining its power resources by forwarding traffic from other nodes, we propose a hierarchical infrastructure where some nodes have more power resources and thus can assist the smaller nodes with communication and data processing. Since the more powerful nodes can offload the smaller nodes, we call the more powerful nodes *proxies*. Our sensor network architecture thus consists of a large number of sensor nodes, a smaller number of proxy nodes, and one or possibly more network sinks.

Often, the proxies can be situated in existing infrastructure. For instance, there are infrastructure networks built in hospitals and industrial factories that could be used to prolong the lifetime of the sensor networks. The infrastructure network can act as a fault tolerant proxy backbone for sensor nodes collecting data or monitoring patients. Industrial and hospital infrastructure networks are static and they do not have limited energy as sensor nodes have. In this paper we assume that the proxies are stationary. Sensor nodes in the network connected to machines, medical equipment, patients etc. have a varying degree of mobility, however we will treat them as if they were mobile and as if the topology of the sensor network was frequently changed. The infrastructure network could be wired, wireless or a combination of both. Some sensitive hospital equipment could be disturbed by wireless transmissions so it may not be feasible to have strong-powered wireless proxies talking to each other. Some of the proxies thus need to be wired and have low-powered wireless transmitters that do not disturb sensitive equipment.

The advantage of using proxies as masters for a sensor *cluster* is that proxies have a lot of memory, high speed processors, “unlimited” energy etc. A proxy can always have the radio transmitter/receiver active to perform complex optimizations and routing for the sensor nodes. A proxy, in our architecture, has large radio coverage and can potentially accept all the sensor nodes that are receiving the signal to its cluster. To build clusters of sensor nodes to reduce the amount of traffic in the network is proposed in e.g. [4]. Some sensor nodes become cluster-heads and collect all traffic from/to their cluster. A cluster-head sends the collected traffic to a gateway in the cluster who will forward the traffic towards the sink.

The most power-consuming activity of a sensor node is typically communication [5]. Communication must hence be kept to a minimum. This applies to transmission, reception and listening for data. All activities involving communication are power-consuming and the most important way to save power is to turn the radio off as much as possible. We therefore propose the use of Time-Division Multiple Access (TDMA) schemes for sensor node communication.

8.1.2 TDMA scheduling for sensor networks

Several different TDMA schemes have been proposed for sensor networks and most of the schemes use sensor nodes to schedule the network.

In [6], methods for reducing energy consumption at all levels of the hierarchy is presented. The sensor nodes communicate with an adjacent basestation within ten meters from the sensor nodes. The sensor nodes send data directly to the basestation without involving other sensor nodes. A sensor node is assumed to synchronize its clock with the basestation several times per second when TDMA is used. When frequency-division multi access (FDMA) is used, the radio will be on for longer periods of time than with TDMA since transmission times are prolonged when using FDMA. FDMA on the other hand does not need to have the sensor nodes' clocks synchronized as TDMA does. The authors of [6] use a hybrid of TDMA and FDMA called TDM/FDM and they give an analytical formula to calculate the optimum number of channels to use in order to get the lowest power consumption.

LEACH is a TDMA cluster-based approach [4]. A node elects itself to be cluster-head by some probability. It broadcasts an advertisement message to the all the other nodes. A none-cluster-head node selects a cluster-head to join by the received signal strength. To be cluster-head is much more energy consuming than to be a non-cluster-head node. All nodes in the network are supposed to be cluster-head during some time period. The TDMA scheme starts every cycle with a set-up phase. After the set-up phase the steady-state phase begins for a certain amount of time. In the steady-state phase there are several frames where nodes have their slots periodically. Then after a certain amount of time the TDMA cycle ends and re-enters the set-up phase.

Dynamically changing the topology without global knowledge of the topology is energy consuming. It is impossible to do optimal route decisions without knowledge of the future topology. Further, several messages have to be exchanged between the sensor nodes to establish and maintain the topology.

In passive clustering [7], no extra messages for building the topology are needed. The first node sending a message will piggyback the sender state to the others. The nodes will form clusters by piggybacking two bits in the MAC layer. A node will need to store cluster-heads and gateways in its memory. If a cluster-head has been silent for a certain amount of time it is removed from the memory. When all the cluster-heads have been removed from the memory, the sensor node will set its state to the initial state and start over again. In [8], the authors extend the passive clustering with a low energy state. Sensor nodes below a certain amount of energy will put themselves in low energy state and will only participate in local collection of data. Still, sensor nodes will need to save the topology in memory and they will need to handle the changes. Also, a cluster-head or a gateway will remain in the same state until the energy falls below a certain threshold.

8.2 Problem Areas

Below we list some important issues.

- As already mentioned in Section 8.1, sensor nodes have scarce resources. A major part of their total energy is used by the wireless radio to send and receive data [5]. It is of great importance to reduce the traffic between sensor nodes in order to prolong their lifetime. Some sensor networks adjust the radio power to save energy. Some networks build clusters, fusion data etc., to reduce the amount of traffic in the network. To organize and distribute the clusters is costly and some sensor nodes will be more exposed than others. The need to reorganize the cluster to spread out the extra workload requires message exchange.
- Sensor nodes could be scheduled or schedule themselves to turn off their radio (sleep) for a specified amount of time. When scheduling themselves to sleep they have to inform the adjacent sensor nodes about this. Sending messages is costly and the energy saved by sleeping could be lost in messages scheduling sensor nodes to sleep.
- Sensor networks using the cluster-based approach could use carrier sense multiple access (CSMA), FDMA, TDMA etc., to schedule the sensor nodes. The radio needs to be turned on frequently when using CSMA. Otherwise it could miss messages from adjacent sensor nodes. Messages from sensor nodes could interfere with each other and result in retransmission of messages.
- Sensor nodes in a TDMA network need to have their clocks well synchronized. Since the clocks of sensor nodes with separate (local) time sources will drift in relation to each other and cause a *clock skew*, sensor node clocks must be resynchronized at regular intervals. If the clocks not are synchronized, scheduled messages could be missed or messages from one sensor node could collide with other messages, i.e., waste of energy. However,

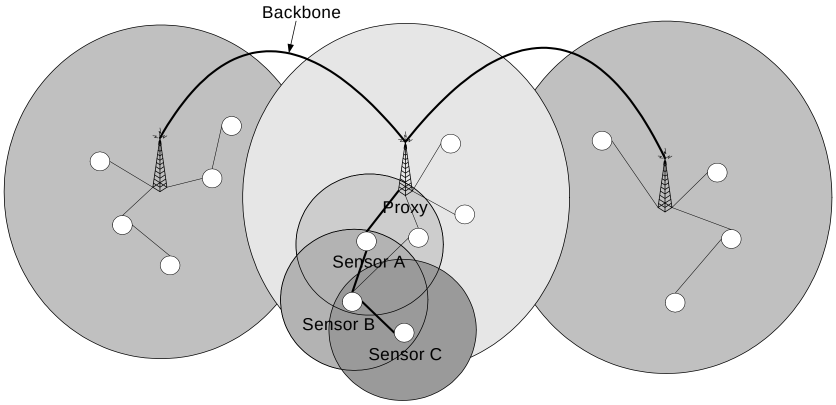

Figure 8.1: Overview of the architecture.

extra messages will have to be exchanged between sensor nodes to keep a global time.

- Routes for messages from a sensor node to the sink will need to be established. Sensor nodes could be added, or disappear forcing new routes to be established. Building routes requires knowledge of the network or message exchange between sensor nodes. Building optimal routes for the packets in the network requires global knowledge of the network architecture. Global knowledge of the network requires a lot of memory, but sensor nodes have a limited amount of memory to their disposition. Using the greater part of the memory to store information about the topology drastically reduces the amount of work a sensor could perform.

  Sensor networks using messages to establish routes by flooding the network, omniscient multicast, advertising/requesting [9, 10] etc., will consume large amounts of energy to establish and maintain routes. Hence, the number of such maintenance messages needs to be minimized.
- To have sensor nodes with different quality of service (QoS) requirements in the same network will increase complexity, computations and radio uptime if managed locally at the sensor nodes. Some optimizations will not be cost-effective in a sensor network, i.e., it would cost more to calculate and distribute the optimization than what could be gained from the optimization itself.

8.3 Using Proxies in Sensor Networks

We propose to build our topology based on clusters with a proxy backbone that has “unlimited” energy and “enough” bandwidth in the backbone channels, see Figure 8.1. The proxy backbone could be e.g. regular computers, PDAs or small embedded systems. The proxies are connected to each other by wire, wirelessly or both. To be able to turn off the radio of the sensor nodes as much as possible, we propose to use TDMA to schedule the communication of the sensor nodes. Furthermore, we propose to build clusters of the sensor nodes where the proxy is the cluster-head. Using clusters will ease the scheduling of the sensor nodes. One proxy is used for each sensor cluster and the proxy is master for the sensor nodes in the cluster. The proxy can reach all the sensor nodes in the cluster directly and a similar TDMA scheme as in LEACH could be used in our topology.

Not all sensor nodes are assumed to be able to communicate directly with the proxy. Some sensor nodes need other sensor nodes to forward the traffic to the proxy. For example, regard Sensor B in Figure 8.1. It is located on the fringe area of the cluster and its radio power is not able to reach the proxy directly. Sensor B needs to use Sensor A to forward its traffic. Sensor B has in its turn to help Sensor C with forwarding of traffic. Thus, we propose an asymmetric topology where the proxy reaches all the sensor nodes in its cluster but the sensor nodes might not reach the proxy directly. This will result in a network hierarchy where proxies are at the top and sensor nodes are divided into different lower levels depending on the sensor nodes’ task etc. Simulations and future experiments will show how to organize the best hierarchy.

The proxy will do the route decisions and manage topology changes for the sensor nodes. A proxy will make a TDMA schedule for its cluster and inform each sensor node about their assigned time slot. The proxy will look at other proxies’ schedules and ensure that its sensor nodes do not interfere with other clusters. The sensor nodes only need to focus on their own tasks and thereby save energy that otherwise would be used to do extra computation and to exchange messages with other sensor nodes in order to maintain the network topology. The proxy will change existing routes to save highly exposed sensor nodes from draining their batteries. When a proxy receives a message from a new non-adjacent sensor node, it will compute the best proxy for that sensor node. The proxy will compute the best route for new sensor nodes and inform the concerned sensor nodes about the changes. It will also check if rearranging old routes to new ones would benefit the sensor nodes. No, or little, knowledge of the network is needed at the sensor nodes, and the memory can be used for data aggregation etc. Proxies can make optimizations that a pure sensor node network would not consider cost-effective by changing the relative cost of the optimization as work is moved from the sensor nodes to the proxy. Issues to solve include

- Mobility: Mobile sensor nodes will make the scheduling decisions worse.
- Energy: When is it worth to reroute the sensor nodes in order to save energy?
- Optimization: What is an optimization goal and when do we execute them?
- New sensor nodes/dead sensor nodes: When to do rerouting and optimizations when a new node enters the cluster or dies?

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 synchronization events 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 cluster could be scheduled for optimized energy saving, others could be scheduled for QoS. In our architecture we can handle sensor nodes with different demands without involving the whole sensor network for reorganization etc. Proxies will handle all extra workload and only the concerned sensor nodes will have to be reorganized. Depending on the application running on the sensor node, i.e. the requested QoS, the proxy 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 sink. Different QoS demands in the network imply high complexity not trivial to solve. We need to group sensor nodes within a cluster in a smart way to guarantee response time etc.

Sensor nodes in a cluster need to help new sensor nodes with connecting to the proxy. A new sensor node will try to contact the closest proxy but sometimes a message could be received by another proxy depending on which sensor node heard the message first. If the new sensor node was heard by an adjacent cluster they will forward the message to its proxy. The proxies then handle the possible handover. Timing issues for the sensor nodes are important to solve. How many cycles after the first request to join a cluster can a new sensor node be guaranteed to be in the cluster? Could a sensor node count on other sensor nodes forwarding the message to the proxy? These questions need to be solved.

The proxy backbone needs to be fault tolerant and if a proxy disappears, other proxies have to take over the orphan cluster. New proxies might enter the backbone and the clusters must be optimized to the new network structure. We need to solve how to handle the re-clustering of the clusters in the network if a proxy should be added, removed or disappear. We need to have solutions for the case if a proxy disappears and the remaining proxies in the network do not reach all the sensor nodes. Traditional sensor schemes could be one way to solve the problem with unreachable sensor nodes.

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] C. Chong and S.P. Kumar. Sensor Networks: Evolution, Opportunities, and Challenges. *Proc IEEE*, August 2003.

[3] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next Centuru Challenges: Scalable Coordination in Sensor Networks. *ACM MobiCOM '99*, 1999.

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

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

[6] E. Shih, S.H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks. *ACM SIGMOBILE*, July 2001.

[7] 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.

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

[9] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. Seattle, WA, 1999. ACM MobiCom '99.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. *ACM MobiCOM '00*, pages 56–67, 2000.

Chapter 9

Paper B
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.

9.1 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 9.1.

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-

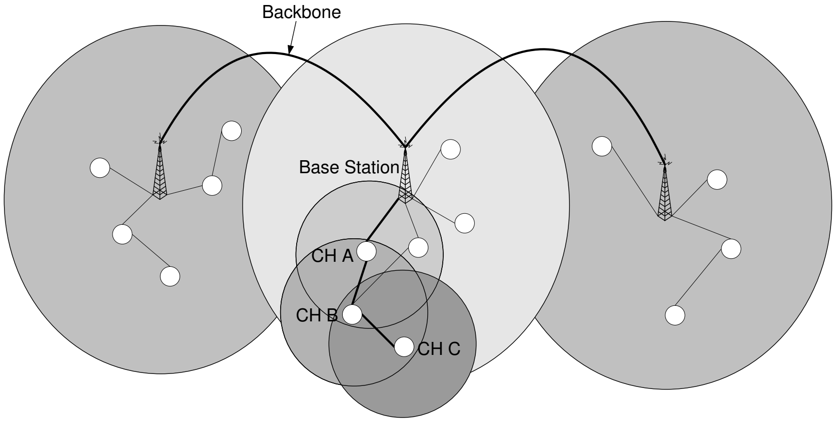

Figure 9.1: Overview of the architecture.

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 9.2, we describe related work. In Section 9.3, the AROS architecture is presented. Section 9.4 describes the comparisons between AROS and the LEACH protocols, and Section 9.5 presents the results from the comparisons. Finally, we conclude and outline future work.

9.2 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 life time. 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.

9.3 AROS

AROS is based on clusters with a Base Station (BS) with “unlimited” energy and “enough” bandwidth in the backbone channels, see Figure 9.1. 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 9.1 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.

9.4 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 9.2. 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 9.1: 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:

$$E_{Tx} = \begin{cases} b * E_{elec} + b * \epsilon_{friss-amp} * d^2 & : d < 87m \\ b * E_{elec} + b * \epsilon_{two-ray-amp} * d^4 & : d \geq 87m \end{cases} \tag{9.1}$$

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

$$E_{Rx} = b * E_{elec} \tag{9.2}$$

Further, all the parameters, such as radio speed, processing delay and radio propagation speed were the same as in [2], see Table 9.1. The energy model can benefit from improvements, however this is outside the scope of this paper.

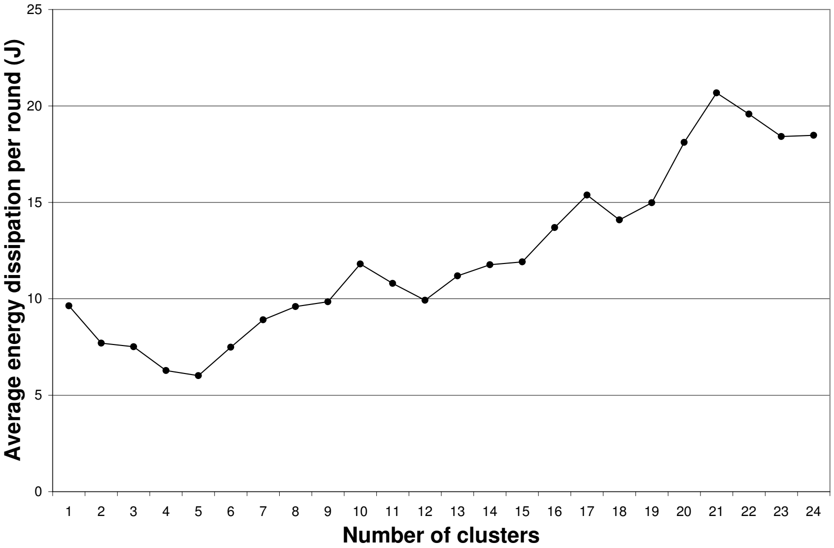

Figure 9.2: Simulation results showing the average energy dissipation per TDMA-round in LEACH.

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 9.2. 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 9.2. All the parameters, except the BS' location and the network size, are the same as in the first simulation setup, see Table 9.1.

We used LEACH-C's centralized cluster formation algorithm to create the

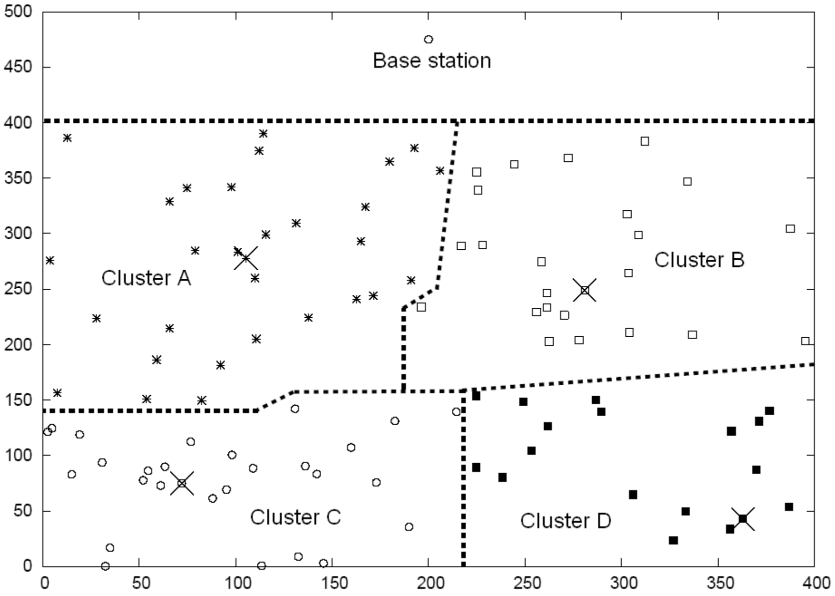

Figure 9.3: Cluster formation of the simulated network using 4 clusters and a network size of 400x400 meters.

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 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 9.3, 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.

9.5 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 9.4. 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 9.4, 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.

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 (9.1) and (9.2) are used to calculate the total energy consumed E_{tot} as:

$$E_{tot} = \begin{cases} E_{Tx_n} & : n = 0 \\ E_{Tx_0} + \sum_{k=1}^{n} (E_{Rx_k} + E_{Tx_k}) & : n > 0 \end{cases} \tag{9.3}$$

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 $E_{tot_{LEACH}} \approx 278mJ, n = 0$ (9.3). The amount of energy consumed when using AROS with

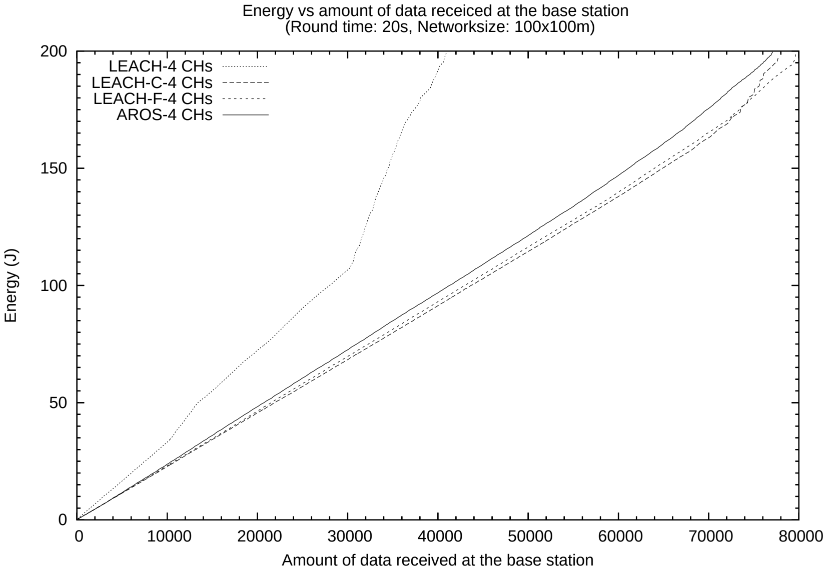

Figure 9.4: Total data received at the base station per given amount of energy in a 100x100 m large network with 4 clusters.

one forwarding CH is $E_{tot_{AROS}} \approx 53mJ$ (9.3). 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.

When comparing how much data the BS receives per Joule of energy in Table 9.2, 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 9.5.

Table 9.2: Data received at base station per unit energy (J)

Protocoll	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	$\frac{37979}{205.2} \approx 185.1$	

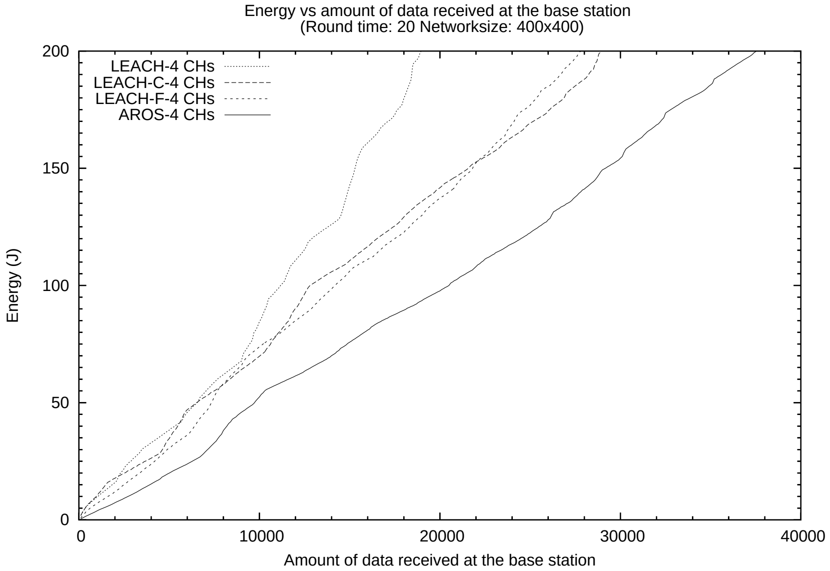

Figure 9.5: Total data received at the base station per given amount of energy in a 400x400 m large network with 4 clusters.

Figure 9.5 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 9.6 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.

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 9.7. 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

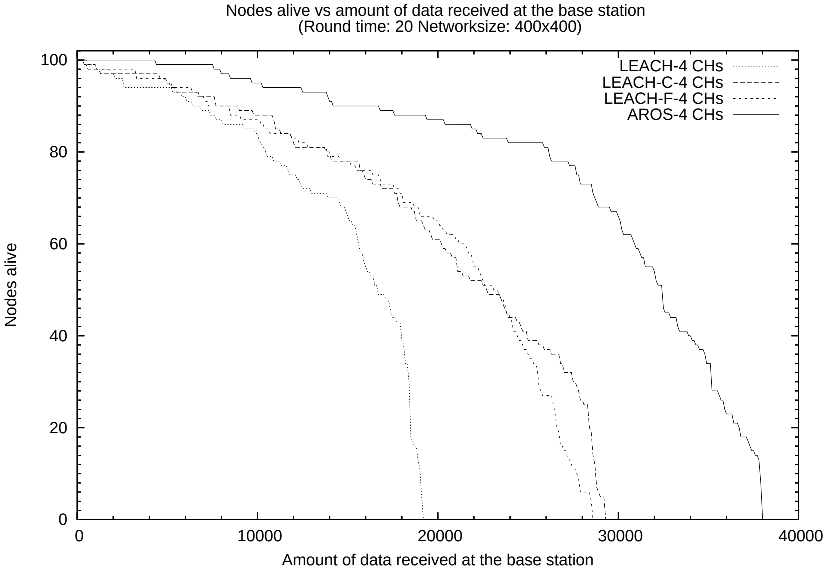

Figure 9.6: Number of nodes alive compared to the amount of data received at the base station in a 400x400 m large network with 4 clusters.

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.

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 9.8. This can also be seen in Figure 9.5, 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.

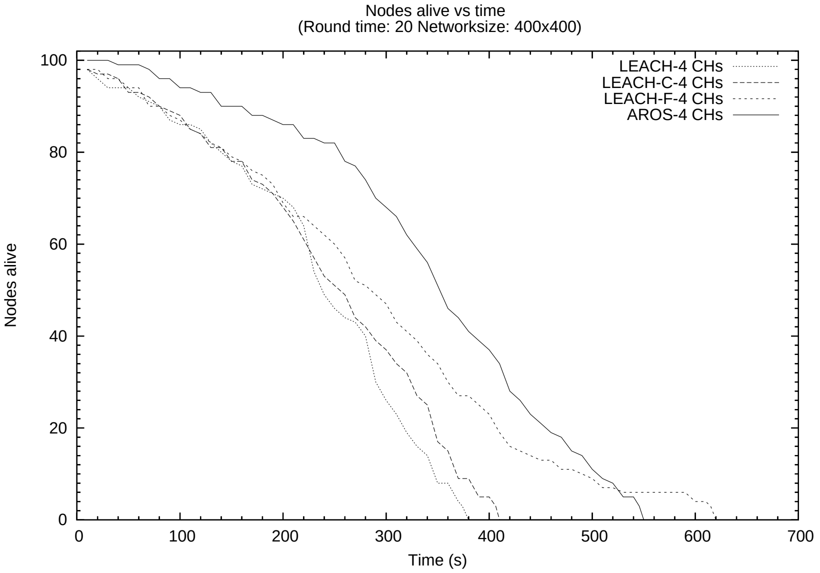

Figure 9.7: The amount of nodes alive over time in a 400x400 m large network with 4 clusters.

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 9.9. 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 life time.

9.6 Conclusions

We have presented a simulation comparison between asymmetric and symmetric communication in sensor networks. In the simulation studies, we have

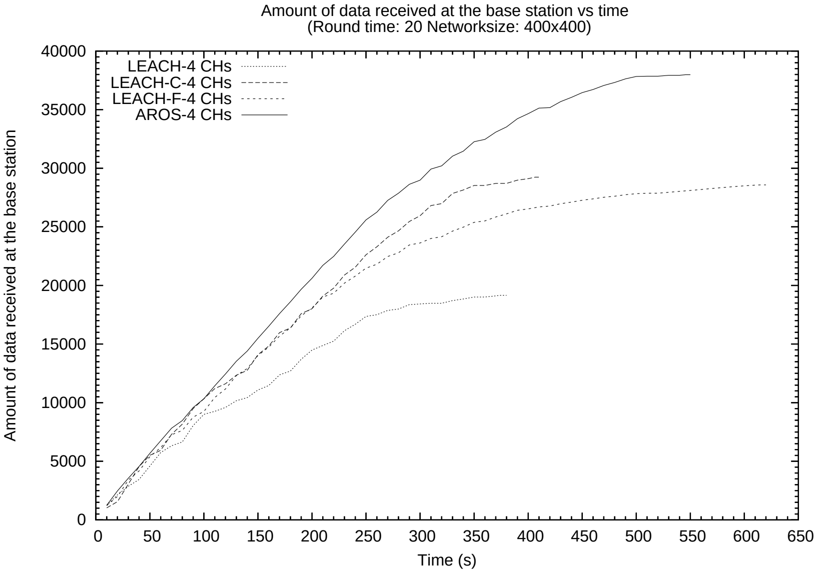

Figure 9.8: Amount of data received at the Base Station over time in a 400x400 m large network with 4 clusters.

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

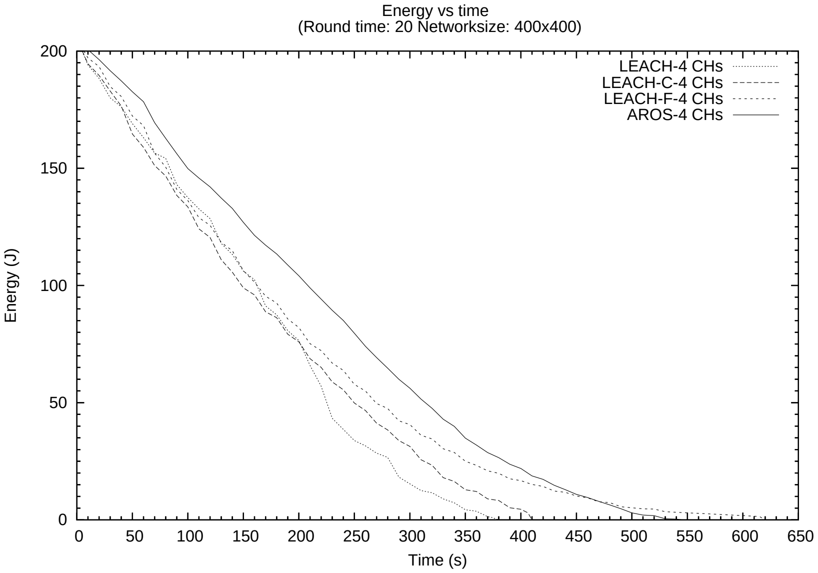

Figure 9.9: Energy left in the system over time in a 400x400 m large network with 4 clusters.

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 10

Paper C: 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

Abstract

In this paper we present a Time Division Multiple Access (TDMA) scheduler for the Asymmetric communication and ROuting in Sensor networks architecture (AROS). The scheduler enables dynamic network configurations of the AROS architecture. We show that asymmetric multihop communication with dynamic network configurations in AROS prolongs the lifetime of sensor nodes in long distance networks compared to the LEACH architecture.

10.1 Introduction

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 supply implies that one important feature of the sensor network is robust functionality in face of network nodes dropping out of the network after some time of activity. Another implication is that, if the network is to survive a longer period of time, new nodes have to be added to the existing network. Thus, the network topology must be dynamic, even if the sensor nodes themselves are not mobile.

In our application areas we like to change all the sensor nodes at one instant in time in order to minimize the maintenance of the network. This implies that the lifetime of the sensor nodes in the network should be as equal as possible, i.e., in the ideal network the sensor nodes would drop out at the same instant in time.

In earlier work [1] we showed that AROS (Asymmetric communication and ROuting in Sensor networks) with a static configuration prolongs the lifetime of long distance networks. The AROS architecture uses the possibility to use asymmetric communication and forwarding of packets [2, 1, 3]. In AROS we use a semi-centralized approach where resource-adequate infrastructure nodes can act as base stations and be used to off-load sensor nodes 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. The infrastructure network can act as a, possibly fault tolerant, base station backbone for sensor nodes collecting data or monitoring of patients.

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 10.1.

In this paper we show that asymmetric communication with a dynamic configuration is better in delivering data to base station than both LEACH [4] and the static configuration of AROS presented in [1]. We present a Time Division Multiple Access (TDMA) scheduler for the AROS architecture, which

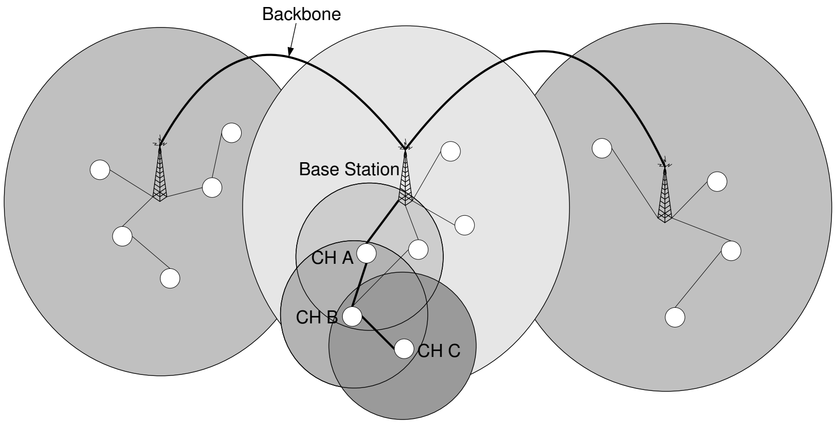

Figure 10.1: Overview of the AROS-architecture.

extends AROS capabilities to handle dynamic network configurations.

The rest of this paper is outlined as follows: in Section 10.2 we describe related work. In Section 10.3, the AROS architecture is presented. Section 10.4 describes the TDMA scheduler and Section 10.5 presents the results from the comparison between the dynamic configuration in AROS and LEACH-C in short and long distance networks. Section 10.5 also presents the results from the comparison between the dynamic simulations and the static simulations made in [1]. In Section 10.6, we conclude the paper and outline some future work.

10.2 Related Work

LEACH (Low-Energy Adaptive Clustering Hierarchy) [4] is a TDMA cluster based approach where a node elects itself to be Cluster Head (CH) by some probability. The sensor nodes create and maintain the network with distributed algorithms. All the sensor nodes in the network have the potential to be CH 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 period of time. The steady-state phases consist of several cycles where all nodes have their slots periodically. The nodes send their data to the CH that aggregates the data and sends it to the 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) [5] has been developed out of LEACH. During the set-up phase, the base station receives information from each node about their current location and energy level. The base station runs the centralized cluster formation algorithm (CCFA) to determine the clusters for that round.

LEACH-F (LEACH with Fixed clusters) [5], is based on clusters that are formed once - and then fixed. The CH position rotates among the nodes within each cluster.

A base station in LEACH-C and LEACH-F has long distance radio coverage and has the potential to accept all the sensor nodes that are receiving the signal from the base station.

10.3 The AROS architecture

The most power-consuming activity of a sensor node is typically radio communication [6]. 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.

One possible solution in order to reduce the amount of traffic in the network is to build clusters of sensor nodes as proposed in, e.g., [7, 4, 8]. Some sensor nodes become CHs and collect all traffic from sensor nodes within the cluster. Furthermore, a CH can also acts as a forwarding node for other CHs. A CH aggregates the collected data from sensor nodes within its cluster, and possibly also the data from other CHs, and then sends that to its Base Station (BS).

AROS is based on clusters with a BS with "unlimited" energy and "enough" bandwidth in the backbone channels. The BSs are connected to each other by wire, wireless, or both. To be able to turn off the radio of the sensor nodes, we 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. When using clusters we can aggregate data to minimize the communication in the network.

A base station in AROS has long distance radio coverage and has the potential to accept all the sensor nodes into its network that are receiving the signal from the base station. The BS can reach all its sensor nodes directly and a similar TDMA scheme as used in LEACH and its variants LEACH-C

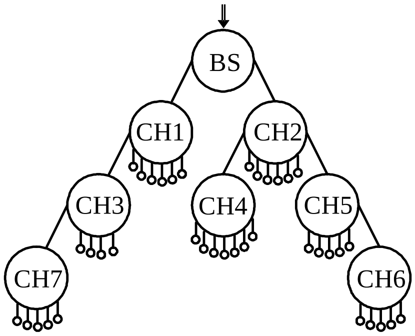

Figure 10.2: A tree example with the relations between the base station and the cluster heads

and LEACH-F could be used in AROS. In AROS, however, asymmetric communication is possible/necessary. 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. Furthermore, for some sensor nodes it may be highly energy consuming to communicate directly with the base station. The traffic from these sensor nodes should rather be forwarded by other sensor nodes in order to save energy. 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.

All clusters in AROS have a CH that aggregates data received from sensor nodes in its cluster. In some applications CHs can aggregate the data received from other CHs, hence reducing the total data size and cycle time. CHs are the only sensor nodes that send and forward data to the BS. As mentioned above, 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 10.1 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 and CHs while the sensor nodes and CHs 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 task 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 new sensor node is assigned, 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. For more information about the BS read [3].

10.4 The AROS TDMA scheduler

In this paper we present a greedy TDMA scheduler for one BS and its sensor nodes. The scheduler enables dynamic network configurations by calculating a new schedule each time the network configuration is changed.

In a network consisting of multiple BSs, each BS can be scheduled in isolation using this algorithm provided that BSs with overlapping radio coverage use separate frequencies. The scheduler can create schedules for networks with or without data aggregation between the CHs. The clusters and the CHs are already chosen before the schedule is created. The schedule is constructed so that a CH does not forward its data until it has received data from all CHs that uses it as a forwarding node. Sensor nodes with different CHs can be scheduled in parallel because they communicate with different frequencies. Further, we schedule CHs sending to different CHs in parallel, using the destination CHs' frequency. Sending the message in parallel will reduce the length of the TDMA cycle, which decreases the delay time for the messages to reach the BS.

10.4.1 Relations between the CHs and the BS

We build a relation tree, based on cluster information, between the sensor nodes and the BS, where the BS is the root node with arbitrary number of CHs as

```
ScheduleNodes($node, slotnumber$)
    $node.slotnumber = slotnumber$
    $children = c(node)$
    While $children \neq \{\}$
        $child = maxc(children)$
        remove child from children
        $slotnumber = slotnumber + 1$
        ScheduleNodes($child, slotnumber$)
```

Figure 10.3: The TDMA schedule algorithm with data aggregation between the CHs

children, see Figure 10.2. The BSs' children can have arbitrary number of CHs as children, see further in Section 10.4.2. The scheduler uses the relation tree to create the TDMA schedule. The relation tree is a partially ordered set with the relation $\succ$ where $x \succ y$ denotes that y is a child to x.

In order to minimize the energy consumption for each individual packet from a CH to the BS, we apply Dijkstra's shortest path algorithm when performing routing decisions, where a path corresponds to energy consumptions.

We use $CH(k)$ to denote that k is a CH, and $shortestPath(CH(k), BS)$ to denote the shortest, most energy efficient, path from CH(k) to the BS.

10.4.2 Scheduling algorithm

In this section we present and describe two different TDMA scheduling algorithms, with and without data aggregation, to enable dynamic network configuration in the AROS architecture. The scheduling algorithms have the goal of minimizing each sensor node's amount of radio uptime as well as minimizing the total schedule length in order to increase the data rate to the BS. To be able to minimize the radio uptime a node should be scheduled to do all of its receiving and sending in adjacent slots.

When a CH aggregates data they receive from its CH descendants, we can safely assume that the CH does not need additional slots in order to forward the data, all received data is aggregated to be sent in one slot. The scheduling algorithm with data aggregation, presented in Figure 10.3, performs a depth-first traversal starting from the BS towards the leafs. This can be seen as the BS sending data downwards in the tree towards the leaves, i.e., a BS to leaves information flow.

At each step it selects the node with the most children first. This means that cycle time can be reduced since sensor nodes within different clusters can be scheduled in parallel. Scheduling a cluster with more sensor nodes before a cluster with fever means that the total length of the combined schedule is shortened.

The resulting total schedule, which now has BS to leaves information flow, should reflect the sending of data from the leaves towards the BS. Therefore, the resulting schedule is reversed in order to get a minimal schedule with leaves to BS information flow.

10.4.2.1 Formal definition of the algorithm

Here we present the formal definition of the algorithm. All the children to a node i is defined as:

$c(i) = \{j | i \succ j\}$

Children, being CHs, to a sensor node i are defined as:

$ch(i) = \{k | k \in c(i) \wedge CH(k)\}$

Children, not being CHs, to a sensor node i are defined as:

$n(i) = \{k | k \in c(i) \wedge \neg CH(k)\}$

All the descendants to a CH are defined as:

$dc(i) = \{j | i \succ j \vee \exists q : i \succ q \wedge j \in dc(q)\}$

$maxc(s)$ returns the node with most children from the set s, and is defined as:

$maxc(s) = k \leftrightarrow \forall k' \in s : |c(k)| \geq |c(k')|$

10.4.2.2 Scheduling example

We use the node topology of Figure 10.2 as a scheduling example. In that example $CH6$ should send the data collected from its cluster nodes to $CH5$. $CH5$ should send the data received from $CH6$ plus its own data collected from its cluster nodes to $CH2$. $CH2$ collects data from its cluster nodes and from $CH5$ and $CH4$ before passing the information to the BS.

Remember that the algorithms start out with scheduling the nodes as the information flows from the BS towards the leaves. Thus, the algorithm starts to schedule CH2 because it has more children than CH1. The algorithm then continues to schedule the CHs at the next level down in the tree, resulting in the leaves of CH4 being scheduled first among all leaves. When all nodes have been scheduled the resulting schedule has to be reversed in order to reflect

Rx \ Tx										
BS									*CH1*	*CH2*
CH1		$N1_1$	$N2_1$	$N3_1$	$N4_1$	$N5_1$	$N6_1$	*CH3*		
CH2		$N1_2$	$N2_2$	$N3_2$	$N4_2$	$N5_2$	$N6_2$	*CH5*	*CH4*	
CH3			$N1_3$	$N2_3$	$N3_3$	$\mathbf{N4_3}$	***CH7***			
CH4	$N1_4$	$N2_4$	$N3_4$	$N4_4$	$N5_4$	$N6_4$	$N7_4$	$N8_4$		
CH5		$N1_5$	$N2_5$	$N3_5$	$N4_5$	$N5_5$	*CH6*			
CH6	$N1_6$	$N2_6$	$N3_6$	$N4_6$	$N5_6$	$N6_6$				
CH7		$N1_7$	$N2_7$	$N3_7$	$N4_7$	$N5_7$				
Slot	1	2	3	4	5	6	7	8	9	10

Figure 10.4: Schedule with data aggregation between the cluster heads

information flow from the leaves to the BS, resulting in the schedule depicted in Figure 10.4.

The schedule in Figure 10.4 shows the receiving nodes (Rx) on the Y axis, the slot number on the X axis and in the grid we see the transmitting nodes (Tx). We see that $CH3$ receives data from its cluster node $N4_3$ at time slot 6 and that $CH3$ receives data from $CH7$ at time slot 7 and so on (highlighted in Figure 10.4).

10.4.2.3 Scheduling algorithm without data aggregation

When data aggregation can not be used, additional slots are needed at the CHs in order to forward the data from other CHs since they can not be aggregated into one message slot. We assume that the data a CH forwards from another CH has to use a whole time slot. Hence, a CH gets as many extra slots as it has CH descendants. The set of CH descendants are defined as:

$dch(i) = \{k | k \in dc(i) \wedge CH(k)\}$

The changes to the previously presented algorithm, for creating a TDMA schedule without data aggregation, are described in Figure 10.5. The presented algorithm needs the following definition:

```
SCHEDULENODES($node, slotnumber$)
    $node.slotnumber = slotnumber$
    $children = c(node)$
    $While\ children \neq \{\}$
        $child = maxca(children)$
        $remove\ child\ from\ children$
        $slotnumber = slotnumber + 1 + |dch(child)|$
        SCHEDULENODES($child, slotnumber$)
```

Figure 10.5: The TDMA schedule algorithm without data aggregation between the CHs

Rx\Tx															
BS									CH1	CH1	CH1	CH2	CH2	CH2	CH2
CH1	$N1_1$	$N2_1$	$N3_1$	$N4_1$	$N5_1$	$N6_1$	CH3	CH3							
CH2			$N1_2$	$N2_2$	$N3_2$	$N4_2$	$N5_2$	$N6_2$	CH5	CH5	CH4				
CH3		$N1_3$	$N2_3$	$N3_3$	$N4_3$	CH7									
CH4			$N1_4$	$N2_4$	$N3_4$	$N4_4$	$N5_4$	$N6_4$	$N7_4$	$N8_4$					
CH5			$N1_5$	$N2_5$	$N3_5$	$N4_5$	$N5_5$	CH6							
CH6		$N1_6$	$N2_6$	$N3_6$	$N4_6$	$N5_6$	$N6_6$								
CH7	$N1_7$	$N2_7$	$N3_7$	$N4_7$	$N5_7$										
Slot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 10.6: Schedule without data aggregation between the cluster heads

$$
\begin{aligned}
&maxca(s) = k \leftrightarrow \\
&\quad \forall k' \in s : |dch(k) \cup n(k)| \geq |dch(k') \cup n(k')|
\end{aligned}
$$

where $maxca(s)$ returns the node with most children and CH descendants from the set s.

The schedule without data aggregation between the CHs will increase the cycle time, hence increase the delay for the BS to receive packets from the sensor nodes. Scheduling the node topology of Figure 10.2 without data aggregation will result in a schedule presented in Figure 10.6.

10.5 Simulations

In [1], we presented a simulation study comparing AROS, with a static network configuration, to LEACH. We investigated the number of data packets received at the BS during the lifetime of the sensor network. The simulations revealed that forwarding, i.e., asymmetric communication, reduces the amount of energy for long distance networks.

In this paper we continue the simulation study of a comparison between AROS and LEACH. In these new simulations AROS is extended to cope with a dynamic network configurations enabled by the presented TDMA scheduler.

10.5.1 Simulation setup

The simulations are performed in NS 2 [9] using the MIT uAMPS ns code extensions [10]. As in [1], the cluster formations are created with the CCFA that LEACH-C uses, see Section 10.2. The BS does not make any optimizations such as e.g., recalculation of the best cluster formation or the optimal sleep time. We assume that the sensor nodes are clock synchronized, and that the position of the sensor nodes can be obtained by the BS.

First, the simulations were configured as in [5] i.e., a network size of 100x100 meters with 100 nodes randomly distributed and the BS 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. 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-C [5]. The energy consumption of the radio transmitter is according to [5] $\epsilon_{friss-amp} = 10pJ/bit/m^2$ 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:

$$E_{Tx} = \begin{cases} b * E_{elec} + b * \epsilon_{friss-amp} * d^2 & : d < 87m \\ b * E_{elec} + b * \epsilon_{two-ray-amp} * d^4 & : d \geq 87m \end{cases} \tag{10.1}$$

Table 10.1: Characteristics of the network

	1:st simu	2:nd sim
Network size	$100X100$ m	$400X400$ m
BS location, x,y	$50, 175$	$200, 475$
Nodes	100	100
Radio prop. speed	3x10^8 m/s	3x10^8 m/s
Processing delay	$50\ \mu$s	$50\ \mu$s
Radio speed	1 Mbps	1 Mbps
Data size	500 bytes	500 bytes

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

$$E_{Rx} = b * E_{elec} \tag{10.2}$$

Further, all the parameters, such as radio speed, processing delay and radio propagation speed were the same as in [5], see Table 10.1. The energy model can benefit from improvements but is outside the scope of this paper.

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 $x = 200$, $y = 475$. All the parameters, except the BS' location and the network size, are the same as in the first simulation setup, see Table 10.1.

10.5.2 Simulation results

In this section we present results from simulations performed in NS 2 with dynamic network configuration enabled by the new TDMA scheduler. The evaluation metric is, as in [1], number of data packets received by the BS during the network life time. All the simulations have been performed without data aggregation between the CHs. If AROS would use data aggregation it would prolong the lifetime of the sensor network even further since the number of slots the CHs use to forward are reduced to one. Thus, in such a simulation AROS would perform even better compared to LEACH.

We start in Section 10.5.2.1 by showing simulations made in a 100x100 meter network, i.e., the same scenario as the original simulations by LEACH-C [5]. In section 10.5.2.2 we increase the network size to 400x400 meters, showing simulation results for a long distance network. We show that AROS with dynamic cluster formations and CHs extends the lifetime of the network, compared to LEACH and its variants, with respect to the amount of energy consumed by the sensor node per data packet sent to the BS.

10.5.2.1 Simulations in a 100x100 meter network

In [1] we showed that AROS performed almost as well as LEACH-C in a 100x100 meter scenario with static clustering, see Figure 10.7. The figure shows the number of nodes alive at the Y-axis and the number of messages received by the BS on the X-axis. The figure plots the three different LEACH variants and AROS, both with static and dynamic configuration.

We can deduce that AROS with dynamic clustering performs as well or better than LEACH-C. AROS chooses the most energy-efficient route to the BS, and if the best route is to send the data directly to the BS then AROS does that, i.e., acts like the LEACH-C protocol. The reason why AROS did not perform as well as LEACH in [1] was that the sensor nodes did not check if the data would reach the BS at the last cycle of each round. When a new round starts every sensor node in the network empty their buffers and wait for the BS to send out their new assignments. Hence, if the sensor nodes do not check if their data reaches the BS at the last cycle of each round, we lose data and waste energy. Today the sensor nodes only schedule themselves to send to its CH if all the sensor nodes in the path to the BS find time to send their own and forward others' data before the round time ends.

From Figure 10.7, we can also discern that AROS, with static configuration, did not perform as well as LEACH-F and LEACH-C. AROS does not perform as well as LEACH-C and LEACH-F due to data losses in the network, as explained above. When comparing AROS with dynamic configuration and data check against the static configuration (without data check), the amount of data received by the BS is increased with approx. 11%, from 77100 to 85700 data packets.

10.5.2.2 Simulations in a 400x400 meter network

In [1], we showed that LEACH-C did not perform well when the network was increased to 400x400 meters. The sensor nodes furthest away from the BS demise early due to the long transmission distances. In all the simulations made with LEACH-C we can see that the sensor nodes furthest away from the BS demise first.

As seen in Figure 10.8, AROS with dynamic configuration delivers more

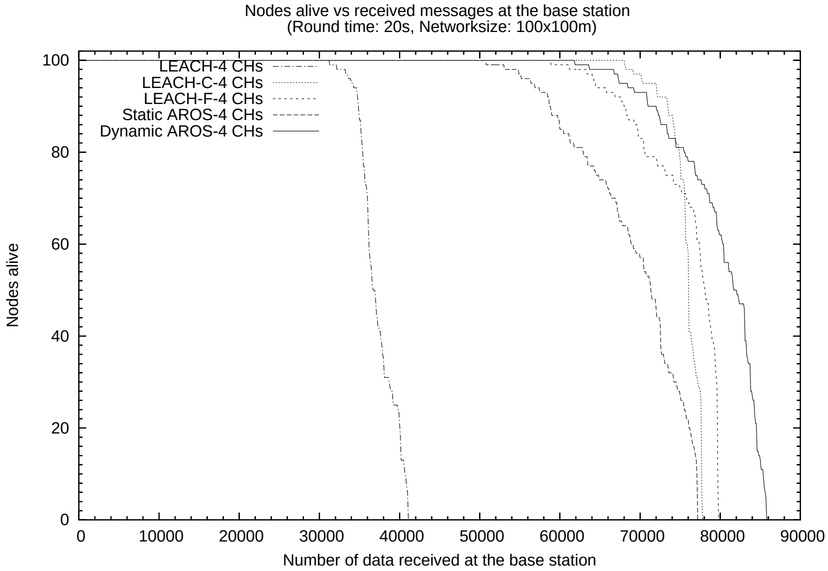

Figure 10.7: Simulation results from the simulations in a 100x100 meter network

messages to the BS than LEACH, LEACH-C and LEACH-F in a 400x400 meter network. AROS delivers 12200 (64%) more messages to the BS than LEACH, 2800 (10%) more messages than LEACH-F and 2100 (7%) more messages than LEACH-C.

In the static simulations made in [1], we showed preliminary results of AROS delivering more messages to the BS in long distance networks than LEACH-C. Simulations with 4 clusters show that CCFA often puts three CHs closely grouped at the back of the network with one CH in the front of the network. This increases the distance a sensor node need to send its data to its CH. Furthermore, the CH in the front of the network need to forward data from all the CHs in the back, hence more energy is consumed than would be done if the clusters are spread across the network. This can be one reason why the static configuration performs better than the dynamic configuration, as seen in Figure 10.8.

In the simulation with static configuration, AROS with static configuration delivers approx. 6600 (21%) more data packets to the BS compared to the dy-

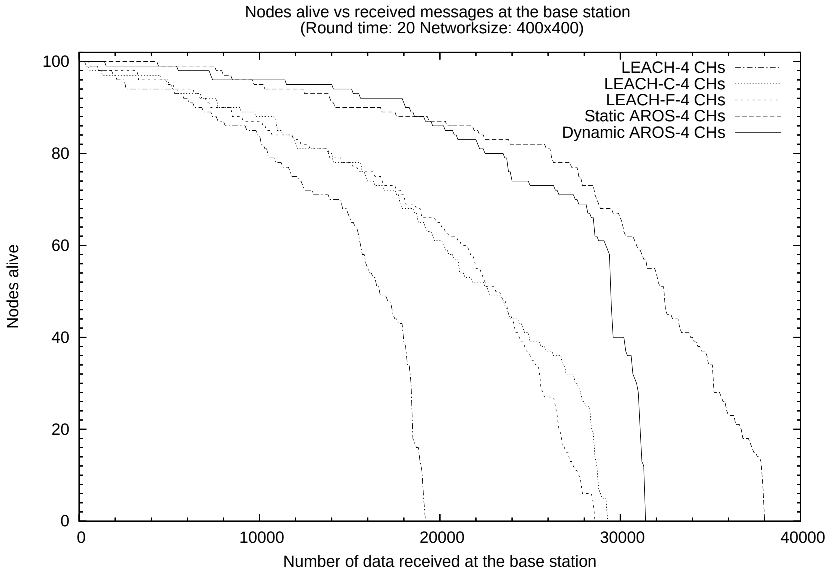

Figure 10.8: Simulation results from the simulations in a 400x400 meter network

namic configuration. We believe that separating the CHs evenly and the use of dynamic clustering will increase the performance even further. By distributing the CHs better in the network, the network could change so that the sensor nodes demise evenly over the network. One possible way to do this is to is to place several CHs in the front of the network and fewer and fewer CHs towards the back of the network.

Having more CHs in the front of the network will share the work of forwarding data from CHs at the back of the network. Work to achieve efficient CH distribution is ongoing. Another reason why the dynamic configuration performs worse could be when several CHs share the same path or parts of a path to the BS. This adds extra workload to those CHs in between the sending CH and the BS. The current algorithm does not take in account that other CHs already might use the path or parts of the path when it creates the shortest path from a CH to the BS, we will extend the algorithm to handle this in future work.

10.6 Conclusions and future work

In this paper we have presented a TDMA scheduler for the AROS architecture enabling dynamic network configurations. We have shown that asymmetric multihop communications with the TDMA scheduler prolongs the lifetime of the sensor nodes with dynamic network configurations in long distance networks.

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. AROS is similar to LEACH-C, a cluster-based protocol where the clusters have CHs that can aggregate and fuse data received from the sensor nodes in its cluster.

In our simulations we have studied how dynamic network clustering in AROS, with non-mobile nodes, affects the amount of data received by the BS. We have shown that AROS is better than LEACH-C in collecting data to a base station with the same total amount of energy for long distance networks and that AROS performs as well or better than LEACH-C in small networks.

We are planning to perform thorough simulations of AROS where we lift some of the restrictions placed on AROS in order to compare it against LEACH. Two such important restriction is 4 CHs and the 20s round time. Our belief is that AROS can perform even better when being able to change the number of CHs and being able to vary round times. Also, the result can be improved when distributing the CHs more evenly over the network. Furthermore, we will investigate other parameters than the number of packets received at the BS. An example result metric include how network life-time is correlated to the delay time in the network. Another important metric is to investigate the lifetime of the sensor nodes. The lifetime should be as equal as possible and in the application areas considered it is preferred to replace all sensor nodes at one instant in time.

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. 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.

[3] 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.

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

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

[7] 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.

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

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

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