A TDMA scheduler for the AROS architecture

Jonas Neander, Ewa Hansen, Jukka Mäki-Turja, Mikael Nolin and Mats Björkman

Mälardalen Real-Time Research Centre

Mälardalen University, Sweden

jonas.neander@mdh.se

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.

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.

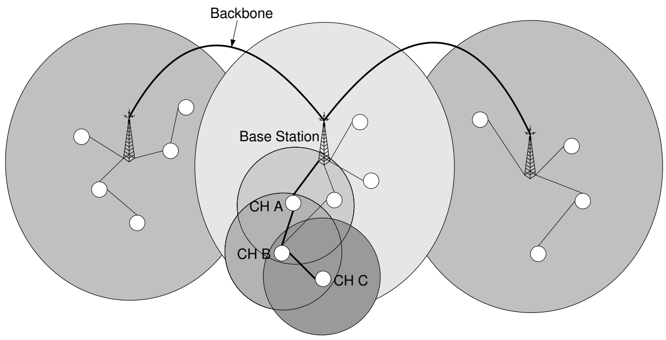

Figure 1. Overview of the AROS-architecture.

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. 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 [6] 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 [5, 6, 7]. 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 1.

In this paper we show that asymmetric communication with a dynamic configuration is better in delivering data to base station than both LEACH [3] and the static configuration of AROS presented in [6]. We present a Time Division Multiple Access (TDMA) scheduler for the AROS architecture, which extends AROS capabilities to handle dynamic network configurations.

The rest of this paper is outlined as follows: in Section 2 we describe related work. In Section 3, the AROS architecture is presented. Section 4 describes the TDMA scheduler and Section 5 presents the results from the comparison between the dynamic configuration in AROS and LEACH-C in short and long distance networks. Section 5 also presents the results from the comparison between the dynamic simulations and the static simulations made in [6]. In Section 6, we conclude the paper and outline some future work.

2. Related Work

LEACH (Low-Energy Adaptive Clustering Hierarchy) [3] 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) [2] 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) [2], 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.

3. The AROS architecture

The most power-consuming activity of a sensor node is typically radio communication [10]. 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., [1, 3, 9]. 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 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

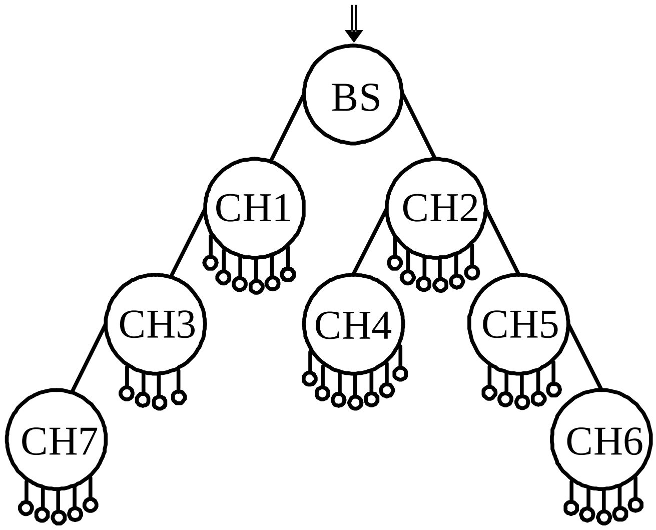

Figure 2. A tree example with the relations between the base station and the cluster heads

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 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 [7].

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.

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 children, see Figure 2. The BSs' children can have arbitrary number of CHs as children, see further in Section 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.

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

```
SCHEDULENODES(node, slotnumber)
  node.slotnumber = slotnumber
  children = c(node)
  While children ≠ {}
    child = maxc(children)
    remove child from children
    slotnumber = slotnumber + 1
    SCHEDULENODES(child, slotnumber)
```

Figure 3. The TDMA schedule algorithm with data aggregation between the CHs

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 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.

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:

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 4. Schedule with data aggregation between the cluster heads

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

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:

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

4.2.2. Scheduling example

We use the node topology of Figure 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

```
ScheduleNodes(node, slotnumber)
    node.slotnumber = slotnumber
    children = c(node)
    While children ≠ {}
        child = maxca(children)
        remove child from children
        slotnumber = slotnumber + 1 + |dch(child)|
        ScheduleNodes(child, slotnumber)
```

Figure 5. The TDMA schedule algorithm without data aggregation between the CHs

reflect information flow from the leaves to the BS, resulting in the schedule depicted in Figure 4.

The schedule in Figure 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 4).

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 5. The presented algorithm needs the following definition:

$maxca(s) = k \leftrightarrow$

$\forall k' \in s : |dch(k) \cup n(k)| \geq |dch(k') \cup n(k')|$

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

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 6. Schedule without data aggregation between the cluster heads

for the BS to receive packets from the sensor nodes. Scheduling the node topology of Figure 2 without data aggregation will result in a schedule presented in Figure 6.

5. Simulations

In [6], 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.

5.1. Simulation setup

The simulations are performed in NS 2 [8] using the MIT uAMPS ns code extensions [4]. As in [6], the cluster formations are created with the CCFA that LEACH-C uses, see Section 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 [2] 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

Table 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 μs	50 μs
Radio speed	1 Mbps	1 Mbps
Data size	500 bytes	500 bytes

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 [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:

$$E_{Tx} = \begin{cases} b * E_{elec} + b * \epsilon_{friss-amp} * d^2 & : d < 87m \\ b * E_{elec} + b * \epsilon_{two-ray-amp} * d^4 & : d \geq 87m \end{cases} \tag{1}$$

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

$$E_{Rx} = b * E_{elec} \tag{2}$$

Further, all the parameters, such as radio speed, processing delay and radio propagation speed were the same as in [2], see Table 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 1.

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 [6], 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 5.2.1 by showing simulations made in a 100x100 meter network, i.e., the same scenario as the original simulations by LEACH-C [2]. In section 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.

5.2.1. Simulations in a 100x100 meter network

In [6] we showed that AROS performed almost as well as LEACH-C in a 100x100 meter scenario with static clustering, see Figure 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 [6] 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 7, we can also discern that AROS, with static configuration, did not perform as well as LEACH-F

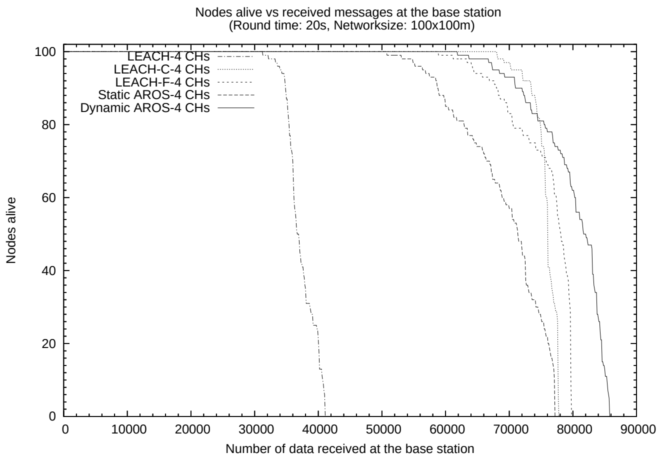

Figure 7. Simulation results from the simulations in a 100x100 meter network

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.

5.2.2. Simulations in a 400x400 meter network

In [6], 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 8, AROS with dynamic configuration delivers more 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 [6], 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,

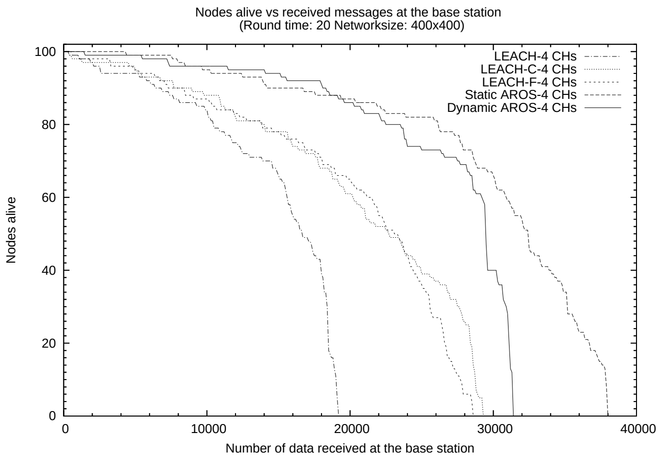

Figure 8. Simulation results from the simulations in a 400x400 meter network

as seen in Figure 8.

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

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.

References

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

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

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

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

[5] 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.

[6] 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.

[7] 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.

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

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

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